Predictive Modeling to Identify Patients Needing a Biopsy to Determine Presence of Cervical Cancer

Shailja Somani & Ruddy Simonpour

University of San Diego

ADS 503-02: Applied Predictive Modeling

June 9, 2022

Abstract

Cervical cancer, a significant global health issue, is a malignant tumor that develops in the cells of the cervix. Early detection and prediction are vital for effective treatment and improved patient outcomes. Our study presents a novel application of data science methodologies for cervical cancer prediction, aiming to contribute to these early detection strategies. Utilizing a comprehensive dataset with 36 variables, we employed both linear and non-linear predictive models including Neural Network (NNET), Multivariate Adaptive Regression Splines (MARS), Support Vector Machine (svmRadial, svmPoly), K-Nearest Neighbour (KNN), Random Forest (RF), Logistic Regression (LR), Linear Discriminant Analysis (LDA), Penalized Logistic Regression, and Nearest Shrunken Centroids, respectively. The target variable, 'Biopsy', is binary, indicating the presence (1) or absence (0) of cancer based on biopsy results. To optimize computational efficiency and model performance, we implemented a dimensionality reduction technique, 'nearZeroVariance'. The models were evaluated using the Area Under the Receiver Operating Characteristic Curve (AUC), a robust measure for classification tasks. Our optimal model, Random Forest (RF), demonstrated exceptional predictive power with an AUC value of (0.8809659). Key predictive factors included 'Schiller', 'Citology', 'IUD', and 'Hormonal Contraceptives', among others. This study underscores the potential of data science, specifically when implemented with the R programming language, in enhancing predictive accuracy and contributing to early detection strategies in cervical cancer.

Keywords: Cervical Cancer, Data Science, R Programming, Predictive Modeling, Random Forest, Dimensionality Reduction, AUC, Schiller, Citology, IUD, Hormonal Contraceptives.

Table of Contents

Introduction ……4

Methodology……5

Data Resource……..6

Data Wrangling and Preprocessing……..6

Exploratory Data Analysis (EDA).....7

Data Partitioning……..9

Class Imbalance Exploration……...9

Scaling and Centring……10

Modeling……...11

Modeling and Evaluation……..11

Recursive Feature Elimination……..12

Results……13

Feature Enhancements……..14

References……15

1. Introduction

Cervical cancer is a leading health issue for women worldwide, as it is the "fourth most common cancer among women globally, with an estimated 604,000 new cases and 342,000 deaths in 2020" (World Health Organization, 2022). In order to adequately catch cervical cancer early and treat it properly, women should be regularly screened for HPV starting "from 30 years of age" (World Health Organization, 2022). Unfortunately, cervical cancer affects women rather young as it is "most often diagnosed between the ages of 35 and 44" (American Society of Clinical Oncology, 2023). Younger women often do not think they are at high risk for cancers, so they fail to get adequately screened or even attend annual exams, which often leads to cancer not getting detected early enough for optimal treatment. As it currently stands, not much progress is being made in impacting patient outcomes in this arena as the cervical cancer "death rate has been declining by less than 1% each year since the early 2000s" (American Society of Clinical Oncology, 2023). Thus, the goal of our work is to push the needle on improving health outcomes for women with cervical cancer by identifying high-risk women who healthcare providers should reach out to for earlier screening and biopsies, as needed. Due to the nature of cervical cancer, in that it has clear correlations with a patient's HPV status, sexual history, pregnancy history, and more, we are able to create a predictive model that uses a woman's medical history to predict if she should obtain a biopsy to determine presence of cervical cancer or not. This paper outlines the methods we used to build such a model.

2. Methodology

Our research on cervical cancer biopsy prediction followed a rigorous and systematic methodology, ensuring the reliability and validity of our findings. The methodology was divided into several key stages, each contributing to the development of our predictive model.

1. Exploratory Data Analysis (EDA): We began with a comprehensive EDA, utilizing both graphical and non-graphical representations to understand the relationships between the response variable and predictor variables. This initial analysis provided valuable insights into the structure and patterns within our dataset.
2. Data Wrangling and Pre-processing: The next stage involved data cleaning and data pre-processing. This stage included handling missing data points, outliers, and dimensionality reduction, ensuring the dataset is ready and well-suited for the modeling process.
3. Data Splitting: To ensure that we have a robust model, we partitioned the dataset into training and validation (test). This approach allowed us to train our models on the training data and then evaluate the performance of models on a separate dataset (test set). This stage is crucial in machine learning to prevent overfitting and to ensure that our models can generalize well to unseen data.
4. Modeling: We defined our main research questions and identified the appropriate analytical methods to address them. This stage involved the selection and implementation of both linear and non-linear models, with a particular focus on non-linear models due to its promising performance in preliminary tests.
5. Validation and Testing: After model implementation, we conducted a thorough validation and performance evaluation. This involved calculating performance metrics across resamples with varying parameters. To enhance accuracy and prevent overfitting, we utilized cross-validation

techniques. This process ensured our models were robust, performing well across different subsets of the dataset, thereby optimizing their performance.

6. Results and Final Model Selection: The final stage involved the analysis of performance measures, leading to the selection of the best model. Our chosen model, Random Forest, represented exceptional predictive power by its high AUC value.

2.1. Data Resource

Our study utilized a dataset sourced from the UCI repository, a renowned platform for machine learning research. This dataset contains 858 instances and 36 variables, providing a comprehensive overview of patients' medical histories from Hospital Universitario de Caracas in Caracas, Venezuela (Fernandes, et al., 2017). The variables encompass a wide range of factors, including the use of hormonal contraceptives and IUDs, the presence of various sexually transmitted diseases (STDs) such as AIDS and HIV, condylomatosis, cervical condylomatosis, and lifestyle activities like smoking and sexual activity. It's important to note that out of the 36 variables, 25 contained missing values, necessitating careful data cleaning and pre-processing.

2.2. Data Wrangling and Pre-processing

Data pre-processing is a pivotal step in any machine learning modeling. In our data wrangling process, we undertook several steps, including checking for missing values, removing variables with degenerate distributions, addressing imbalanced dataset issues, and eliminating outliers. The dataset contained numerous missing data points, with two columns, ‘STDs: time since first diagnosis’ and ‘STDs: time since last diagnosis’, having more than 90% missing values, equating to 787 missing data points. We decided to exclude these columns from the dataset due to the substantial amount of missing information.

Other columns had less than 15% missing values. For these, we opted to impute the missing points using the KnnImpute function. This method works by identifying the k nearest neighbors of each data point with missing values, then imputing the missing values with the mean value of these k nearest neighbors (Brownlee, 2020). In this study, we set K=10 for the KnnImpute algorithm to impute missing data in our dataset. The original dataset comprised 858 observations and 36 variables. However, several variables exhibited degenerate distributions, meaning they had only one possible value. We removed these variables from the dataset using the nearZeroVariance function, resulting in a dataset with 858 observations and 18 variables. After excluding the two columns due to substantial missing values, our final dataset consisted of 858 observations and 16 variables. In addition, we also utilized an analysis to remove highly correlated variables from the dataset. A cut-off 0.9 was chosen for the correlation coefficient, which is generally considered a very strong correlation. Implementing high-correlation cut-off led to the elimination of three variables from the dataset, thereby reducing the total number of variables in the dataset to 13. This pre-processing ensured our dataset was well-structured and suitable for the subsequent stages of our analysis.

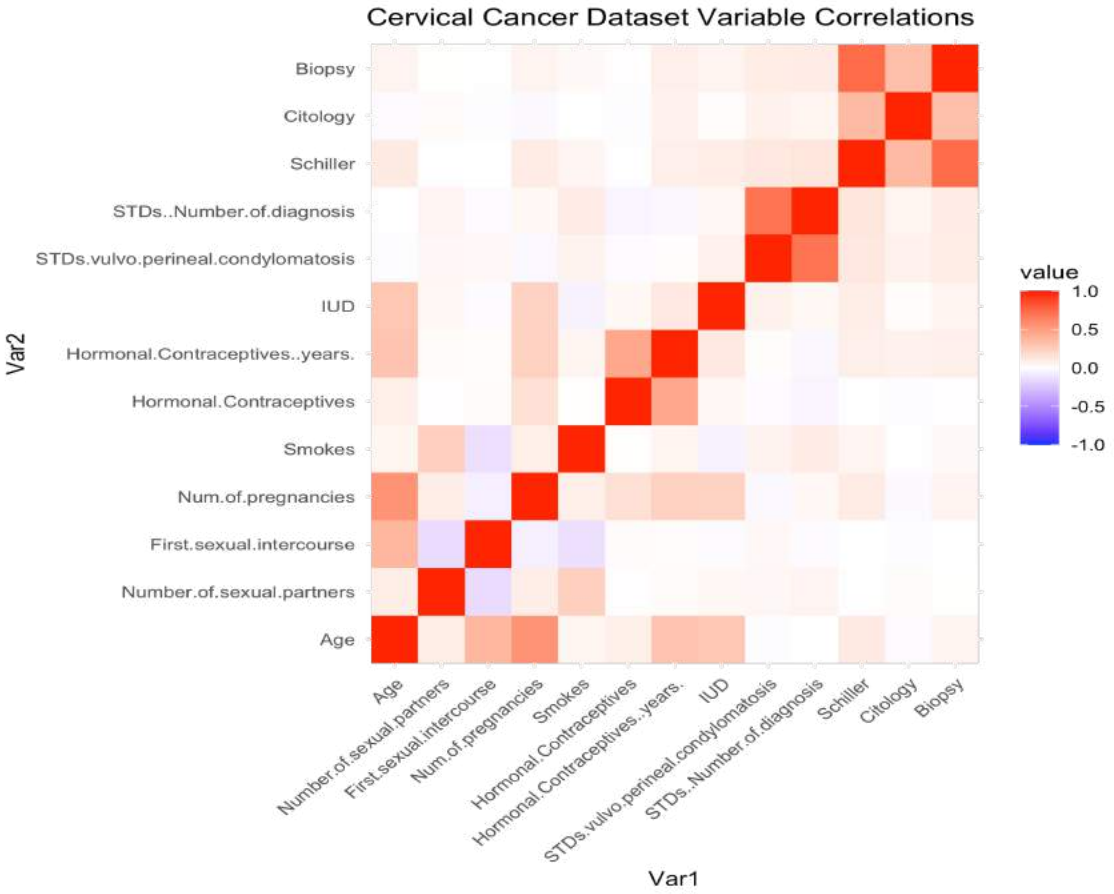
2.3 Exploration Data Analysis (EDA)

We conducted Exploratory Data Analysis (EDA) to understand the relationships and patterns between the variables. One of the key elements to find the association within datasets, is implementing a correlation matrix graph. In our correlation matrix graph, red and blue colors were used to indicate the direction and strength of the relationship. Red cells represent a strong positive correlation, implying that as one variable increases, the other variable also tends to increase. On the other hand, blue cells indicated a negative correlation, suggesting that as one variable increases, the other decreases. The intensity of the colors provided additional

information about the correlation strength. Faded colors suggested weaker correlations, while more intense colors represented stronger correlations.

Figure 1

Correlation matrix graph between variables



2.4. Data Partitioning

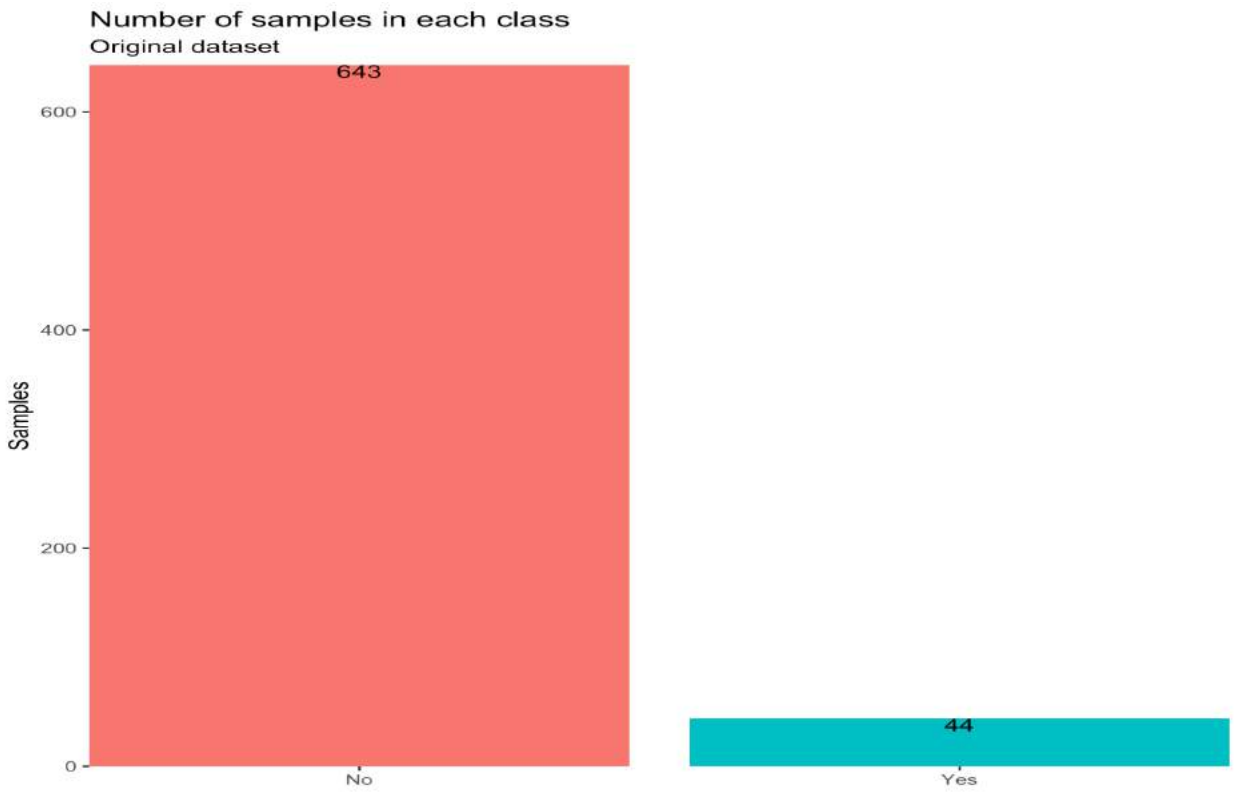
In our study, we partitioned the dataset into two subsets: 80% of the data was allocated to the training set (train_X), and the remaining 20% was reserved for the test set (test_X). This 80-20 split is a common practice in machine learning, providing a good balance between maximizing the amount of training data and ensuring a sufficient amount of data for testing.

2.4.1. Class Imbalance Exploration

Class imbalance is a common challenge in many machine learning projects. In our analysis, we used the 'Biopsy' variable as the response (target) variable. This binary variable is coded as '1' (if a patient had a biopsy and the result was positive for cervical cancer, and '0' if there was no need for a biopsy and no detection of cervical cancer. There are several strategies to address the issue of class imbalance, including under-sampling, random over-sampling (ROSE), and Synthetic Minority Over-sampling Technique (SMOTE), among others. After careful consideration, we decided to employ the ROSE method to balance our dataset. The original distribution of the class imbalance was 643 for '0'/'No' (no cancer) and 44 for '1'/'Yes' (cancer).

Figure 2

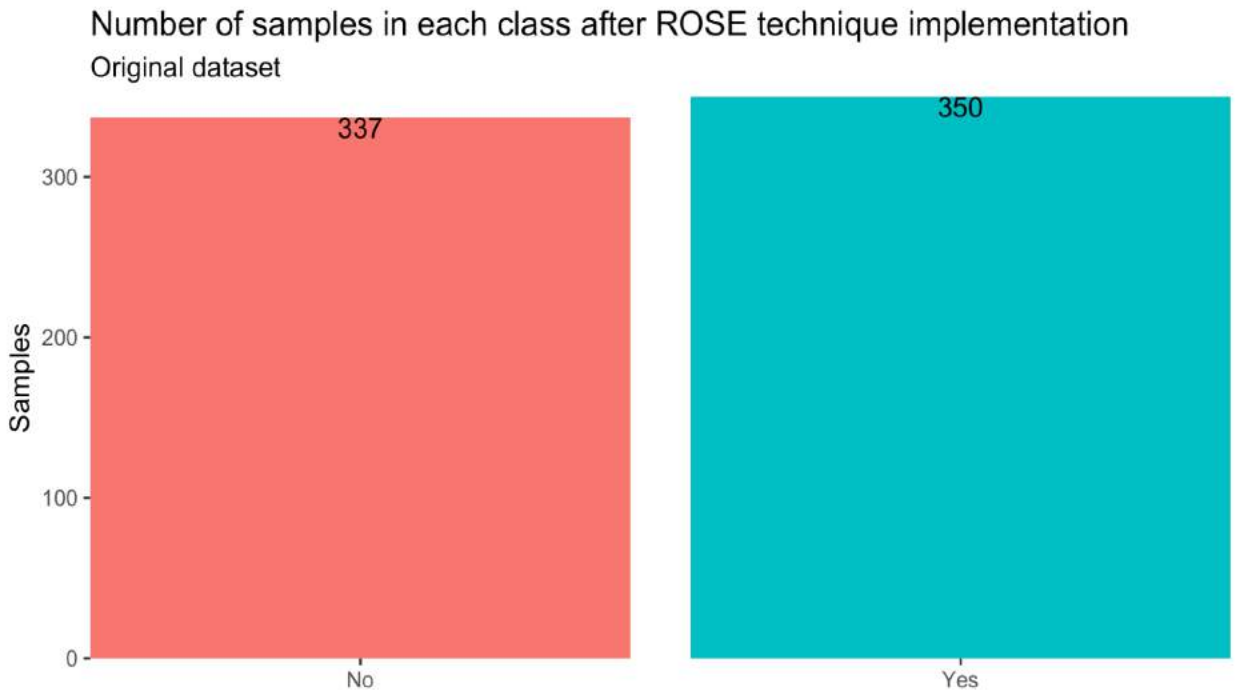
Class Imbalance of training dataset



After employing the ROSE technique, the class distribution was significantly improved. The ROSE technique is an oversampling method that generates artificial samples according to a smoothed bootstrap approach. It works by creating synthetic samples from the feature space neighborhood around the minority class (DemiR & ŞahiN, 2022). This technique is significantly useful when dealing with an imbalanced dataset, it can help to equalize the class distribution and thus improve the accuracy and efficiency of the model. As a result of implementation, the class distribution in our dataset was adjusted to 337 for '0'/'No' and 350 for '1'/'Yes'.

Figure 3

Class Imbalance After ROSE technique Implementation



2.4.2 Scaling and Centering

Following the partitioning and balancing the dataset, we performed data preprocessing on both the training and test sets. This included scaling and centering the predictor variables, which are standard preprocessing steps to ensure that all variables are on a similar scale and have a

mean of zero. Importantly, we computed the scaling and centering parameters (i.e., the mean and standard deviation) using only the training data. These parameters were then applied to both the training and test sets. This approach prevents information leakage from the test set into the training process, maintaining the integrity of our evaluation. It's worth noting that standardizing the response variable is not necessary because the relationship between the predictors and the target doesn't change with scaling. These steps can help improve the performance of many machine learning algorithms by ensuring that all features contribute equally to the model, regardless of their original scales.

2.5 Modeling

2.5.1 Modeling and Evaluation

We investigated a wide variety of linear and nonlinear modeling techniques in order to determine the best algorithm for this prediction problem. In order to have replicable modeling results, we set the seed to 100 within each of our modeling functions. The nonlinear algorithms we applied to this dataset were: (1) Neural Network, (2) Multivariate Adaptive Regression Splines (MARS), (3) Support Vector Machine (SVM) with Radial Basis, (4) SVM with Polynomial Kernel, (5) K-Nearest Neighbors (KNN), and (6) Random Forest (RF). The linear algorithms we applied were: (1) Logistic Regression, (2) Linear Discriminant Analysis (LDA), (3) Penalized Logistic Regression, and (4) Nearest Shrunken Centroids. When applicable, we tuned hyperparameters for models using ten-fold cross-validation to maximize model performance. We then calculated the accuracy, sensitivity, specificity, confusion matrices, and AUC for each model. However, when determining which model had the best performance, we chose AUC as our evaluation metric due to the fact that it incorporates both the false positive and false negative rate rather than considering only one metric. Because we also conducted ROSE to

balance our dataset, we do not have to be as worried about drawbacks of using AUC when working with unbalanced datasets.

2.5.2 Recursive Feature Elimination

Recursive Feature Elimination (RFE) is a popular feature selection technique in machine learning. The core idea behind RFE is to iteratively eliminate the least important features while training a model. Initially, the model is trained on the entire feature set. Then, the importance of each feature is assessed, often by examining the coefficients in regression models, or by measuring the amount a tree-based model's accuracy decreases when the feature is randomly permuted. The least important feature(s) are then removed, and the model is re-trained. This process repeats until the desired number of features is achieved, or further removal of features does not contribute to a more straightforward, yet still accurate, model. RFE is an effective way to deal with high-dimensional data where some features may be irrelevant or redundant. By removing such features, RFE can often improve the model's predictive performance and make the model more interpretable. In this study, the RFE method was applied to the optimal model, Random Forest, to see if eliminating less important features could enhance the performance. However, the AUC score slightly dropped from 0.8803977 (using original features) to 0.8761 (using RFE). There could be several reasons why the RFE function did not improve the model's performance.

1. In some cases, a feature that appears unimportant in isolation might be very important when combined with other features. RFE, which typically considers features individually, might mistakenly eliminate such features.

2. While RFE can help prevent overfitting by reducing model complexity, it is also possible that the original model was not overfitting the data. In such a case, reducing model complexity could actually make the model worse, which might be what happened here.

3. Results

The below table shows the optimal hyperparameters for all models we tried, as well as the resulting AUC for each model (given a seed of 100).

Table 1

Hyperparameter tuning and modeling results

Model	**Tuning Hyperparameter Results**	**AUC**
Neural Network	size = 3; decay = 0.1	0.8661932
Multivariate Adaptive Regression Splines	nprune = 13; degree = 1	0.8389205
SVM with Radial Basis Function Kernel	sigma = 0.05553555; C = 1	0.8647727
SVM with Polynomial Kernel	degree = 2; scale = 0.01; C = 16	0.7977273
K-Nearest Neighbors	k = 1	0.8633523
Random Forest	mtry = 7	0.8803977
Logistic Regression	N/A	0.8056818
Linear Discriminant Analysis	N/A	0.7920455
Penalized Logistic Regression	alpha = 0.1; lambda = 0.01	0.8045455
Nearest Shrunken Centroids	threshold = 1.724138	0.8247159

Among the different models tested, the Random Forest algorithm demonstrated superior performance, with an accuracy of 93%, sensitivity of 95%, specificity of 72%, and an AUC-ROC

of 0.88. These results suggest a high diagnostic accuracy and precision of the Random Forest model in detecting cervical cancer. The most influential features identified were Schiller, STDs.vulvo.perineal.comdylomatosis, number of different STDs diagnosis, Number of pregnancies, number of sexual partners, and smoking status. This aligns well with established risk factors for cervical cancer.

4. Feature Enhancements

While our project has achieved significant strides in the modeling and feature engineering phases, there is still ample opportunity for enhancement, particularly in the utility and effectiveness of machine learning models used for diagnosing cervical cancer. We identify this as a crucial area for future improvements to further advance the field.

Data Diversity: The data used in this study was collected from 858 women at 'Hospital Universitario de Caracas' in Caracas, Venezuela, which could inherently contain biases related to socioeconomic status, geographic location, or access to healthcare. These biases may affect the model's performance when applied to more diverse populations. To mitigate this, future studies should consider collecting data from different resources.

Constant Model Updates: Machine learning models are not static. As new data becomes available, the models should be retrained and updated to ensure they adapt to changes and improvements in cervical cancer diagnostics and treatment.

Feature Expansion: While the features utilized in this contributed significantly to model performance, there are other potential features that can be improved. These elements can be genetic information, lifestyle habits like use of birth control pills, multiple pregnancies, poor diet. By expanding the feature set, we may be able to further enhance the model's performance.

References

Brownlee, J. (2020, June 24). Machine Learning Mastery. KNN Imputation for Missing Values in Machine Learning. Retrieved June 12, 2023, from https://machinelearningmastery.com/knn-imputation-for-missing-values-in-machine-learning/.

Cervical Cancer Fact Sheet. (2022, Feb 22). World Health Organization. Retrieved June 10, 2023, from https://www.who.int/news-room/fact-sheets/detail/cervical-cancer.

Cervical Cancer: Statistics. (2023, Feb). American Society of Clinical Oncology Cancer.Net. Retrieved June 10, 2023, from https://www.cancer.net/cancer-types/cervical-cancer/statistics.

DemiR, S., & ŞahiN, E. K. (2022). Evaluation of Oversampling Methods (OVER, SMOTE, and ROSE) in Classifying Soil Liquefaction Dataset based on SVM, RF, and Naïve Bayes. *European Journal of Science and Technology*. Retrieved June 12, 2023, from https://doi.org/10.31590/ejosat.1077867.

Fernandes, K., Cardoso, J., & Fernandes, J. (2017). *Cervical cancer (Risk Factors)* [Data set]. UCI Machine Learning Repository. Retrieved May 30, 2023, from https://doi.org/10.24432/C5Z310.

ADS 503: Cervical Cancer Biopsy Prediction Project

Ruddy Simonpour & Shailja Somani

May 30, 2023

```
# load necessary packages for files above
library(Hmisc)
```

```
##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:Hmisc':
##
##     src, summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(reshape2)
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```
suppressWarnings({

#setwd("/Users/shailjasomani/Documents/USD_MS_ADS/ADS_503/Final_Proj")   #choose a location/path and se


setwd("/Users/ruddysimonpour/Desktop/University of Sandiego - Curriculum/ADS 503 - Applied Predictive M

source ("Data_Ingestion.R")
source ("Viz_EDA.R")
source ("Preprocessing.R")
source ("Modeling.R")

})
```

```
## Loading required package: colorspace
```

```
##
## Attaching package: 'colorspace'
```

```
## The following object is masked from 'package:pROC':
##
##     coords
```

```
## Loading required package: grid
```

```
## The legacy packages maptools, rgdal, and rgeos, underpinning this package
## will retire shortly. Please refer to R-spatial evolution reports on
## https://r-spatial.org/r/2023/05/15/evolution4.html for details.
## This package is now running under evolution status 0
```

```
## VIM is ready to use.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
```

```
##
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```

Data Importing

```
# Uses functions from files loaded in to clean data
set.seed(007)

# loading Data
cervical_data_raw <- read_data(x="/Users/ruddysimonpour/Desktop/University of Sandiego - Curriculum/ADS
```

```
## Rows: 858
## Columns: 36
## $ Age                                 <int> 18, 15, 34, 52, 46, 42, 51, 26, 45,~
## $ Number.of.sexual.partners           <dbl> 4, 1, 1, 5, 3, 3, 3, 1, 1, 3, 3, 1,~
## $ First.sexual.intercourse            <dbl> 15, 14, NA, 16, 21, 23, 17, 26, 20,~
## $ Num.of.pregnancies                  <dbl> 1, 1, 1, 4, 4, 2, 6, 3, 5, NA, 4, 3~
## $ Smokes                              <dbl> 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0,~
## $ Smokes..years.                      <dbl> 0.000000, 0.000000, 0.000000, 37.00~
## $ Smokes..packs.year.                 <dbl> 0.0, 0.0, 0.0, 37.0, 0.0, 0.0, 3.4,~
## $ Hormonal.Contraceptives             <dbl> 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1,~
## $ Hormonal.Contraceptives..years.     <dbl> 0.00, 0.00, 0.00, 3.00, 15.00, 0.00~
## $ IUD                                 <dbl> 0, 0, 0, 0, 0, 0, 1, 1, 0, NA, 0, 0~
## $ IUD..years.                         <dbl> 0, 0, 0, 0, 0, 0, 7, 7, 0, NA, 0, 0~
## $ STDs                                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs..number.                       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs.condylomatosis                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs.cervical.condylomatosis        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs.vaginal.condylomatosis         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs.vulvo.perineal.condylomatosis <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs.syphilis                       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs.pelvic.inflammatory.disease    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs.genital.herpes                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs.molluscum.contagiosum          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs.AIDS                           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs.HIV                            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs.Hepatitis.B                    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs.HPV                            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs..Number.of.diagnosis           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STDs..Time.since.first.diagnosis    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ STDs..Time.since.last.diagnosis     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ Dx.Cancer                           <int> 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,~
## $ Dx.CIN                              <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Dx.HPV                              <int> 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,~
## $ Dx                                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,~
## $ Hinselmann                          <int> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,~
## $ Schiller                            <int> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,~
## $ Citology                            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Biopsy                              <int> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,~
```

```
#cervical_data_raw <- read_data(x='/Users/shailjasomani/Documents/USD_MS_ADS/ADS_503/Final_Proj/kag_ris
head(cervical_data_raw,5)
```

```
##   Age Number.of.sexual.partners First.sexual.intercourse Num.of.pregnancies
## 1  18                          4                        15                  1
```

```
## 2  15                          1                          14                  1
## 3  34                          1                          NA                  1
## 4  52                          5                          16                  4
## 5  46                          3                          21                  4
##   Smokes Smokes..years. Smokes..packs.year. Hormonal.Contraceptives
## 1      0              0                   0                       0
## 2      0              0                   0                       0
## 3      0              0                   0                       0
## 4      1             37                  37                       1
## 5      0              0                   0                       1
##   Hormonal.Contraceptives..years. IUD IUD..years. STDs STDs..number.
## 1                               0   0           0    0             0
## 2                               0   0           0    0             0
## 3                               0   0           0    0             0
## 4                               3   0           0    0             0
## 5                              15   0           0    0             0
##   STDs.condylomatosis STDs.cervical.condylomatosis STDs.vaginal.condylomatosis
## 1                   0                            0                          0
## 2                   0                            0                          0
## 3                   0                            0                          0
## 4                   0                            0                          0
## 5                   0                            0                          0
##   STDs.vulvo.perineal.condylomatosis STDs.syphilis
## 1                                  0             0
## 2                                  0             0
## 3                                  0             0
## 4                                  0             0
## 5                                  0             0
##   STDs.pelvic.inflammatory.disease STDs.genital.herpes
## 1                                0                   0
## 2                                0                   0
## 3                                0                   0
## 4                                0                   0
## 5                                0                   0
##   STDs.molluscum.contagiosum STDs.AIDS STDs.HIV STDs.Hepatitis.B STDs.HPV
## 1                          0         0        0                0        0
## 2                          0         0        0                0        0
## 3                          0         0        0                0        0
## 4                          0         0        0                0        0
## 5                          0         0        0                0        0
##   STDs..Number.of.diagnosis STDs..Time.since.first.diagnosis
## 1                         0                              NA
## 2                         0                              NA
## 3                         0                              NA
## 4                         0                              NA
## 5                         0                              NA
##   STDs..Time.since.last.diagnosis Dx.Cancer Dx.CIN Dx.HPV Dx Hinselmann
## 1                              NA         0      0      0  0          0
## 2                              NA         0      0      0  0          0
## 3                              NA         0      0      0  0          0
## 4                              NA         1      0      1  0          0
## 5                              NA         0      0      0  0          0
##   Schiller Citology Biopsy
## 1        0        0      0
```

```
## 2        0        0      0
## 3        0        0      0
## 4        0        0      0
## 5        0        0      0
```

```
dim(cervical_data_raw)
```

```
## [1] 858  36
```

```
# check missing data
null_counts_raw <- check_nulls(cervical_data_raw)
```

```
##                                                                  Column
## Age                                                                 Age
## Number.of.sexual.partners                     Number.of.sexual.partners
## First.sexual.intercourse                       First.sexual.intercourse
## Num.of.pregnancies                                 Num.of.pregnancies
## Smokes                                                         Smokes
## Smokes..years.                                         Smokes..years.
## Smokes..packs.year.                                 Smokes..packs.year.
## Hormonal.Contraceptives                       Hormonal.Contraceptives
## Hormonal.Contraceptives..years.       Hormonal.Contraceptives..years.
## IUD                                                               IUD
## IUD..years.                                               IUD..years.
## STDs                                                             STDs
## STDs..number.                                           STDs..number.
## STDs.condylomatosis                               STDs.condylomatosis
## STDs.cervical.condylomatosis             STDs.cervical.condylomatosis
## STDs.vaginal.condylomatosis               STDs.vaginal.condylomatosis
## STDs.vulvo.perineal.condylomatosis STDs.vulvo.perineal.condylomatosis
## STDs.syphilis                                       STDs.syphilis
## STDs.pelvic.inflammatory.disease     STDs.pelvic.inflammatory.disease
## STDs.genital.herpes                               STDs.genital.herpes
## STDs.molluscum.contagiosum                 STDs.molluscum.contagiosum
## STDs.AIDS                                                   STDs.AIDS
## STDs.HIV                                                     STDs.HIV
## STDs.Hepatitis.B                                     STDs.Hepatitis.B
## STDs.HPV                                                     STDs.HPV
## STDs..Number.of.diagnosis                   STDs..Number.of.diagnosis
## STDs..Time.since.first.diagnosis     STDs..Time.since.first.diagnosis
## STDs..Time.since.last.diagnosis       STDs..Time.since.last.diagnosis
## Dx.Cancer                                                   Dx.Cancer
## Dx.CIN                                                         Dx.CIN
## Dx.HPV                                                         Dx.HPV
## Dx                                                                 Dx
## Hinselmann                                                 Hinselmann
## Schiller                                                     Schiller
## Citology                                                     Citology
## Biopsy                                                         Biopsy
## 37                                                              Total
##                                          Nulls      ColumnPercentage
## Age                                          0                     0
## Number.of.sexual.partners                   26      3.03030303030303
```

```
## First.sexual.intercourse                     7     0.815850815850816
## Num.of.pregnancies                          56      6.52680652680653
## Smokes                                      13      1.51515151515152
## Smokes..years.                              13      1.51515151515152
## Smokes..packs.year.                         13      1.51515151515152
## Hormonal.Contraceptives                    108      12.5874125874126
## Hormonal.Contraceptives..years.            108      12.5874125874126
## IUD                                        117      13.6363636363636
## IUD..years.                                117      13.6363636363636
## STDs                                       105      12.2377622377622
## STDs..number.                              105      12.2377622377622
## STDs.condylomatosis                        105      12.2377622377622
## STDs.cervical.condylomatosis               105      12.2377622377622
## STDs.vaginal.condylomatosis                105      12.2377622377622
## STDs.vulvo.perineal.condylomatosis         105      12.2377622377622
## STDs.syphilis                              105      12.2377622377622
## STDs.pelvic.inflammatory.disease           105      12.2377622377622
## STDs.genital.herpes                        105      12.2377622377622
## STDs.molluscum.contagiosum                 105      12.2377622377622
## STDs.AIDS                                  105      12.2377622377622
## STDs.HIV                                   105      12.2377622377622
## STDs.Hepatitis.B                           105      12.2377622377622
## STDs.HPV                                   105      12.2377622377622
## STDs..Number.of.diagnosis                    0                     0
## STDs..Time.since.first.diagnosis           787      91.7249417249417
## STDs..Time.since.last.diagnosis            787      91.7249417249417
## Dx.Cancer                                    0                     0
## Dx.CIN                                       0                     0
## Dx.HPV                                       0                     0
## Dx                                           0                     0
## Hinselmann                                   0                     0
## Schiller                                     0                     0
## Citology                                     0                     0
## Biopsy                                       0                     0
## 37                                 total_nulls total_percentage_null
```

```
# remove cols with more than 85% missing data
cervical_data_clean <- remove_cols(cervical_data_raw)
```

```
##                                                                Column
## Age                                                               Age
## Number.of.sexual.partners                  Number.of.sexual.partners
## First.sexual.intercourse                    First.sexual.intercourse
## Num.of.pregnancies                                Num.of.pregnancies
## Smokes                                                       Smokes
## Smokes..years.                                       Smokes..years.
## Smokes..packs.year.                                Smokes..packs.year.
## Hormonal.Contraceptives                        Hormonal.Contraceptives
## Hormonal.Contraceptives..years.      Hormonal.Contraceptives..years.
## IUD                                                              IUD
## IUD..years.                                             IUD..years.
## STDs                                                            STDs
## STDs..number.                                          STDs..number.
## STDs.condylomatosis                                 STDs.condylomatosis
```

```
## STDs.cervical.condylomatosis             STDs.cervical.condylomatosis
## STDs.vaginal.condylomatosis               STDs.vaginal.condylomatosis
## STDs.vulvo.perineal.condylomatosis STDs.vulvo.perineal.condylomatosis
## STDs.syphilis                                         STDs.syphilis
## STDs.pelvic.inflammatory.disease     STDs.pelvic.inflammatory.disease
## STDs.genital.herpes                               STDs.genital.herpes
## STDs.molluscum.contagiosum                 STDs.molluscum.contagiosum
## STDs.AIDS                                               STDs.AIDS
## STDs.HIV                                                 STDs.HIV
## STDs.Hepatitis.B                                 STDs.Hepatitis.B
## STDs.HPV                                                 STDs.HPV
## STDs..Number.of.diagnosis                 STDs..Number.of.diagnosis
## STDs..Time.since.first.diagnosis     STDs..Time.since.first.diagnosis
## STDs..Time.since.last.diagnosis       STDs..Time.since.last.diagnosis
## Dx.Cancer                                               Dx.Cancer
## Dx.CIN                                                     Dx.CIN
## Dx.HPV                                                     Dx.HPV
## Dx                                                             Dx
## Hinselmann                                             Hinselmann
## Schiller                                                 Schiller
## Citology                                                 Citology
## Biopsy                                                     Biopsy
## 37                                                          Total
##                                        Nulls      ColumnPercentage
## Age                                        0                     0
## Number.of.sexual.partners                 26      3.03030303030303
## First.sexual.intercourse                   7     0.815850815850816
## Num.of.pregnancies                        56      6.52680652680653
## Smokes                                    13      1.51515151515152
## Smokes..years.                            13      1.51515151515152
## Smokes..packs.year.                       13      1.51515151515152
## Hormonal.Contraceptives                  108      12.5874125874126
## Hormonal.Contraceptives..years.          108      12.5874125874126
## IUD                                      117      13.6363636363636
## IUD..years.                              117      13.6363636363636
## STDs                                     105      12.2377622377622
## STDs..number.                            105      12.2377622377622
## STDs.condylomatosis                      105      12.2377622377622
## STDs.cervical.condylomatosis             105      12.2377622377622
## STDs.vaginal.condylomatosis              105      12.2377622377622
## STDs.vulvo.perineal.condylomatosis       105      12.2377622377622
## STDs.syphilis                            105      12.2377622377622
## STDs.pelvic.inflammatory.disease         105      12.2377622377622
## STDs.genital.herpes                      105      12.2377622377622
## STDs.molluscum.contagiosum               105      12.2377622377622
## STDs.AIDS                                105      12.2377622377622
## STDs.HIV                                 105      12.2377622377622
## STDs.Hepatitis.B                         105      12.2377622377622
## STDs.HPV                                 105      12.2377622377622
## STDs..Number.of.diagnosis                  0                     0
## STDs..Time.since.first.diagnosis         787      91.7249417249417
## STDs..Time.since.last.diagnosis          787      91.7249417249417
## Dx.Cancer                                  0                     0
## Dx.CIN                                     0                     0
```

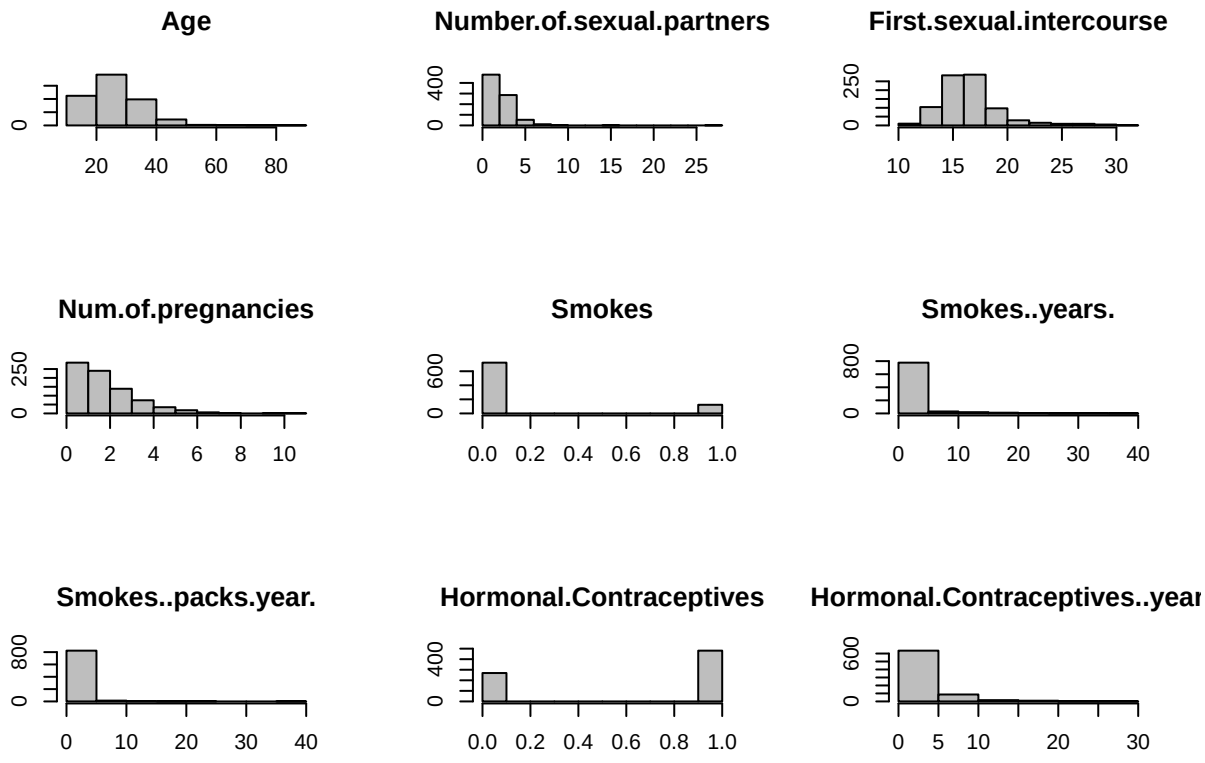
```
## Dx.HPV                                0                    0
## Dx                                    0                    0
## Hinselmann                            0                    0
## Schiller                              0                    0
## Citology                              0                    0
## Biopsy                                0                    0
## 37                        total_nulls total_percentage_null
```

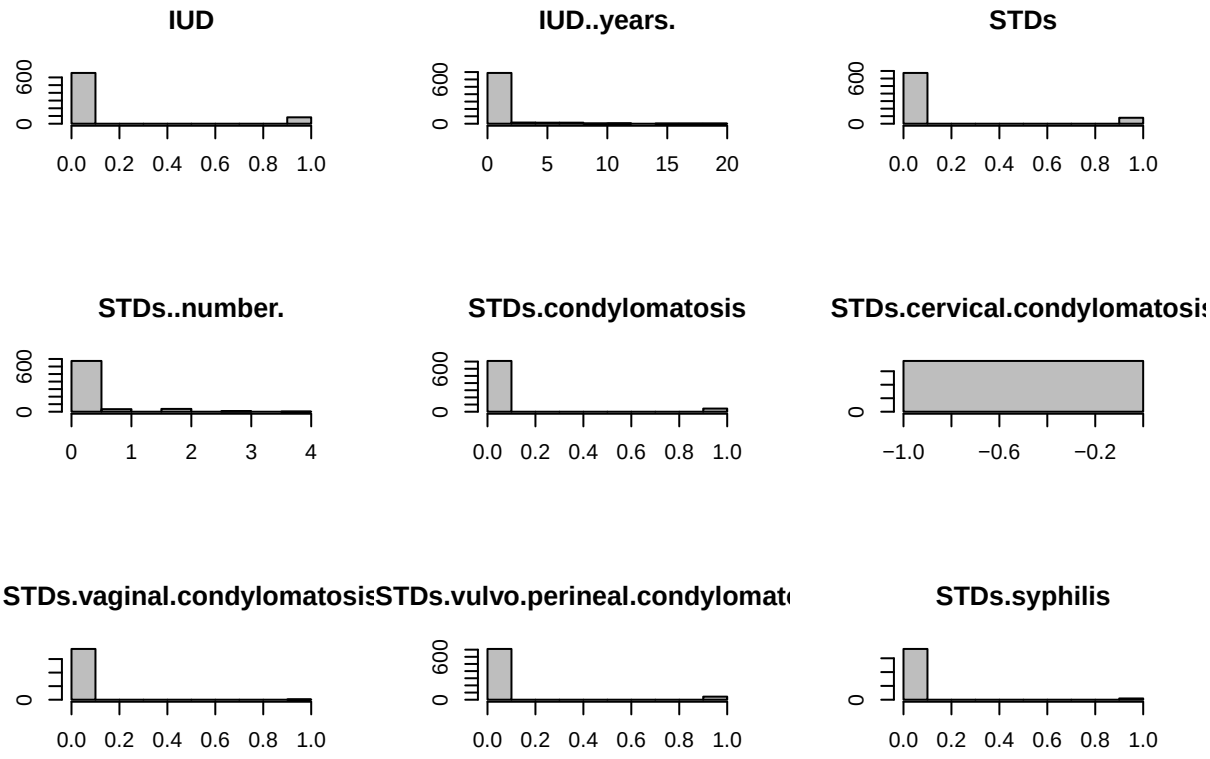
```
dim(cervical_data_clean)
```

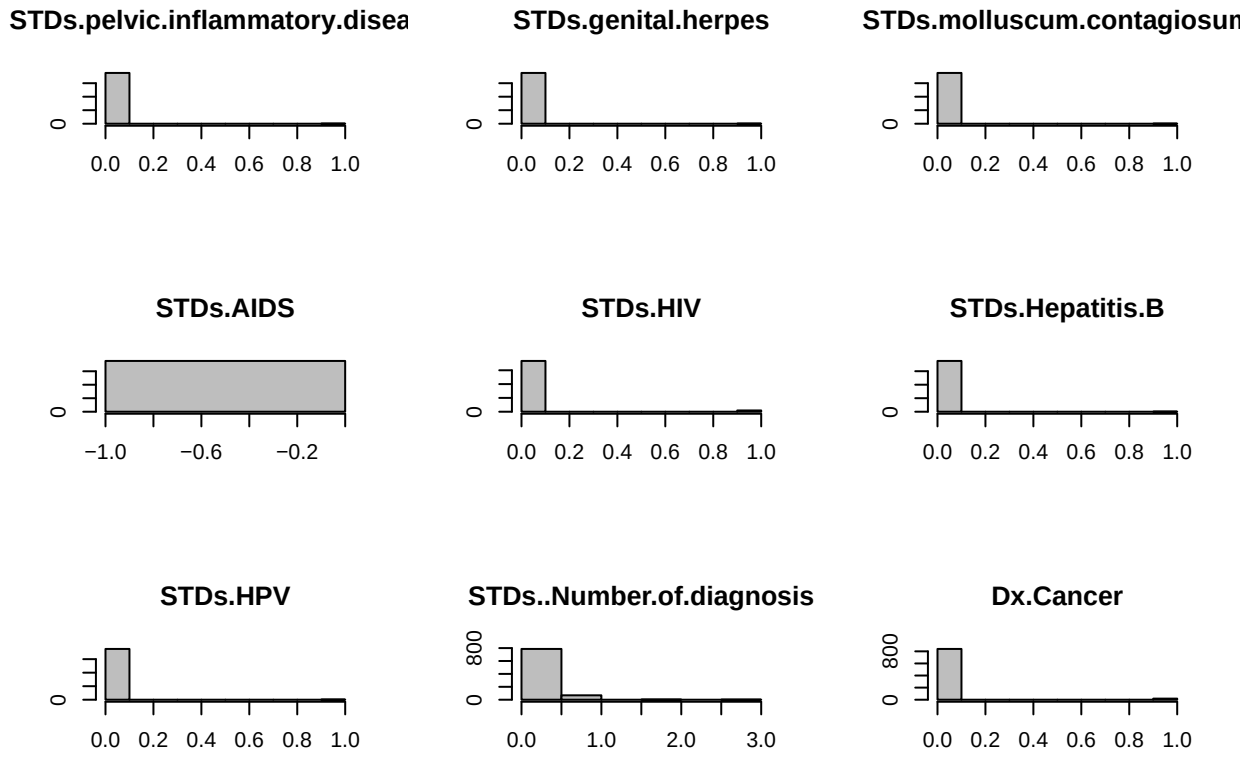
```
## [1] 858  34
```

EDA Analysis

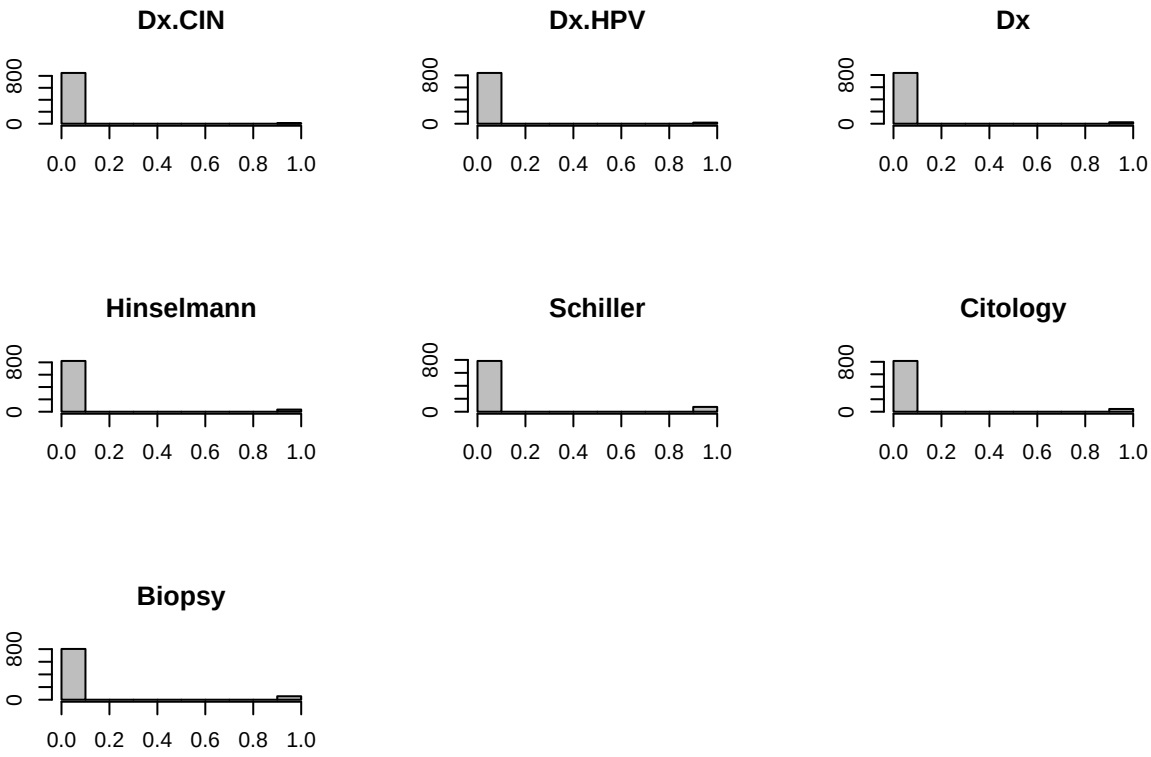
```
# These user-defined functions are pulled from the Viz_EDA.R file.
# Look at all histograms of features collectively
hist.df(cervical_data_clean)
```

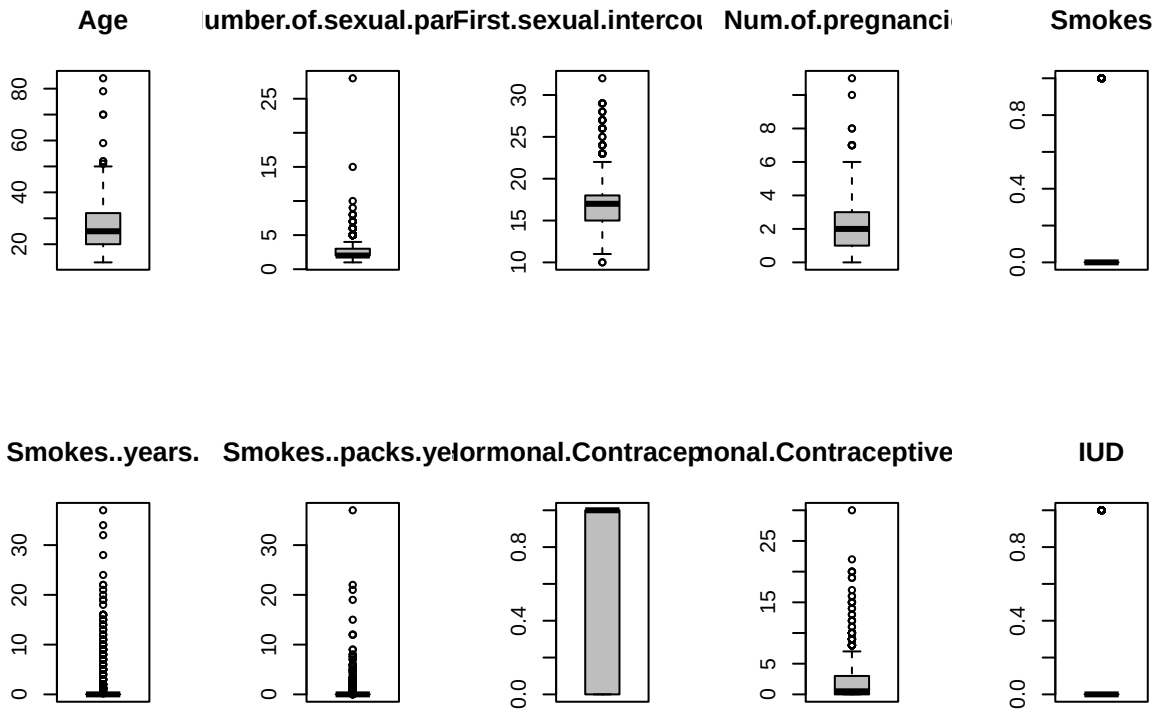


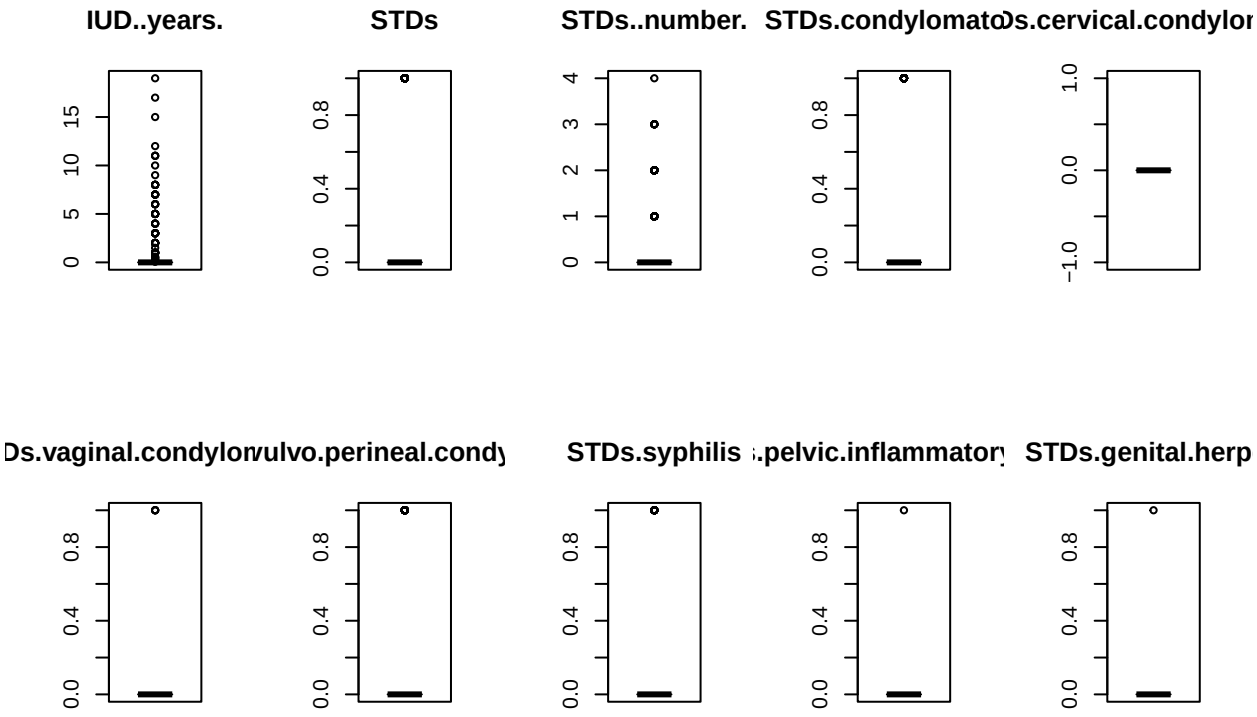

IUD
0 600
0.0 0.2 0.4 0.6 0.8 1.0
IUD..years.
0 600
0 5 10 15 20
STDs
0 600
0.0 0.2 0.4 0.6 0.8 1.0
STDs..number.
0 600
0 1 2 3 4
STDs.condylomatosis
0 600
0.0 0.2 0.4 0.6 0.8 1.0
STDs.cervical.condylomatosis
0
−1.0 −0.6 −0.2
STDs.vaginal.condylomatosis
0
0.0 0.2 0.4 0.6 0.8 1.0
STDs.vulvo.perineal.condylomat
0 600
0.0 0.2 0.4 0.6 0.8 1.0
STDs.syphilis
0
0.0 0.2 0.4 0.6 0.8 1.0

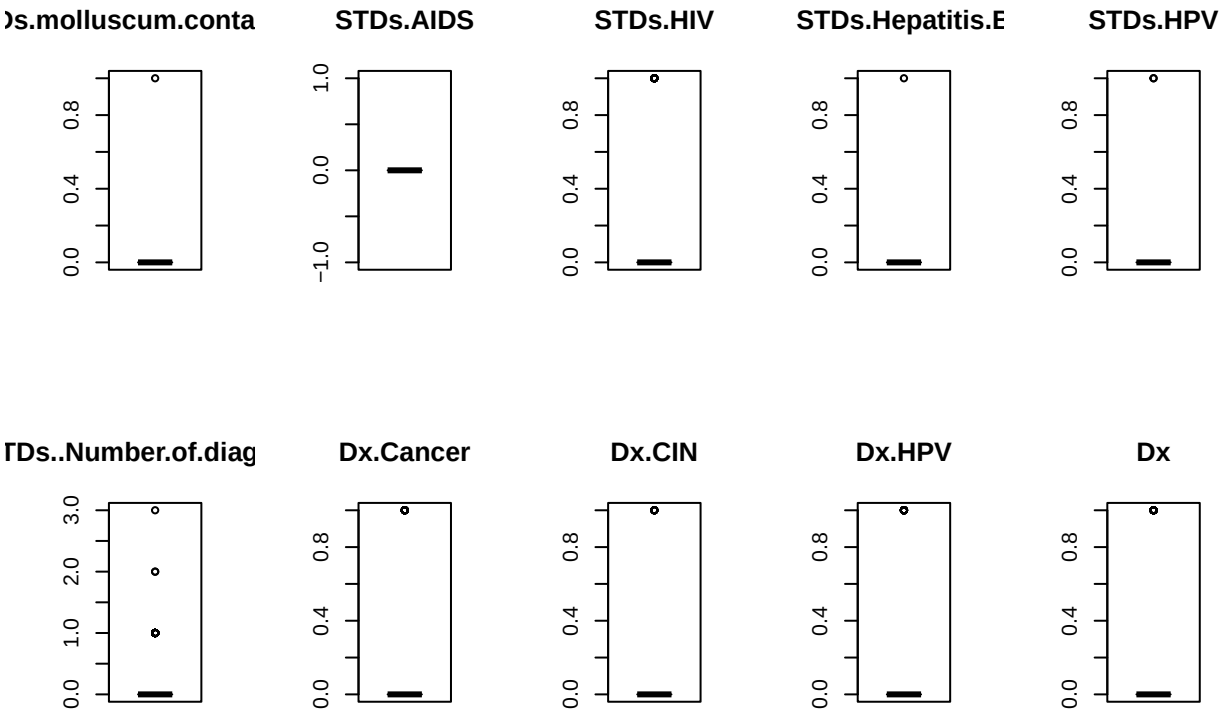


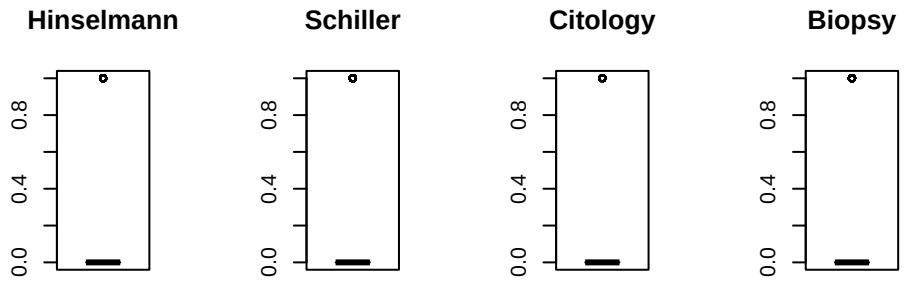

```
# Create boxplots for all features - helps visualize outliers
boxplot.df(cervical_data_clean)
```


Dx.CIN
0
800
0.0 0.2 0.4 0.6 0.8 1.0
Dx.HPV
0
800
0.0 0.2 0.4 0.6 0.8 1.0
Dx
0
800
0.0 0.2 0.4 0.6 0.8 1.0
Hinselmann
0
800
0.0 0.2 0.4 0.6 0.8 1.0
Schiller
0
800
0.0 0.2 0.4 0.6 0.8 1.0
Citology
0
800
0.0 0.2 0.4 0.6 0.8 1.0
Biopsy
0
800
0.0 0.2 0.4 0.6 0.8 1.0


Age
umber.of.sexual.par
First.sexual.interco
Num.of.pregnanci
Smokes
Smokes..years.
Smokes..packs.ye
ormonal.Contracep
nal.Contraceptive
IUD


IUD..years.
STDs
STDs..number.
STDs.condylomato
Ds.cervical.condylor
0 5 10 15
0.0 0.4 0.8
0 1 2 3 4
0.0 0.4 0.8
−1.0 0.0 1.0
Ds.vaginal.condylor
vulvo.perineal.condy
STDs.syphilis
.pelvic.inflammator
STDs.genital.herp
0.0 0.4 0.8
0.0 0.4 0.8
0.0 0.4 0.8
0.0 0.4 0.8
0.0 0.4 0.8


Os.molluscum.conta
STDs.AIDS
STDs.HIV
STDs.Hepatitis.E
STDs.HPV
0.0
0.4
0.8
−1.0
0.0
1.0
0.0
1.0
2.0
3.0
TDs..Number.of.diag
Dx.Cancer
Dx.CIN
Dx.HPV
Dx



Data Cleaning

```
library(caret)

# remove near zero variance variables
dim(cervical_data_clean)
```

```
## [1] 858  34
```

```
degeneratecols <- nearZeroVar(cervical_data_clean)

length(degeneratecols) # number of cols that are degenerate distributions
```

```
## [1] 18
```

```
cervical_data_process <- cervical_data_clean[, -degeneratecols]
dim(cervical_data_process)
```

```
## [1] 858  16
```

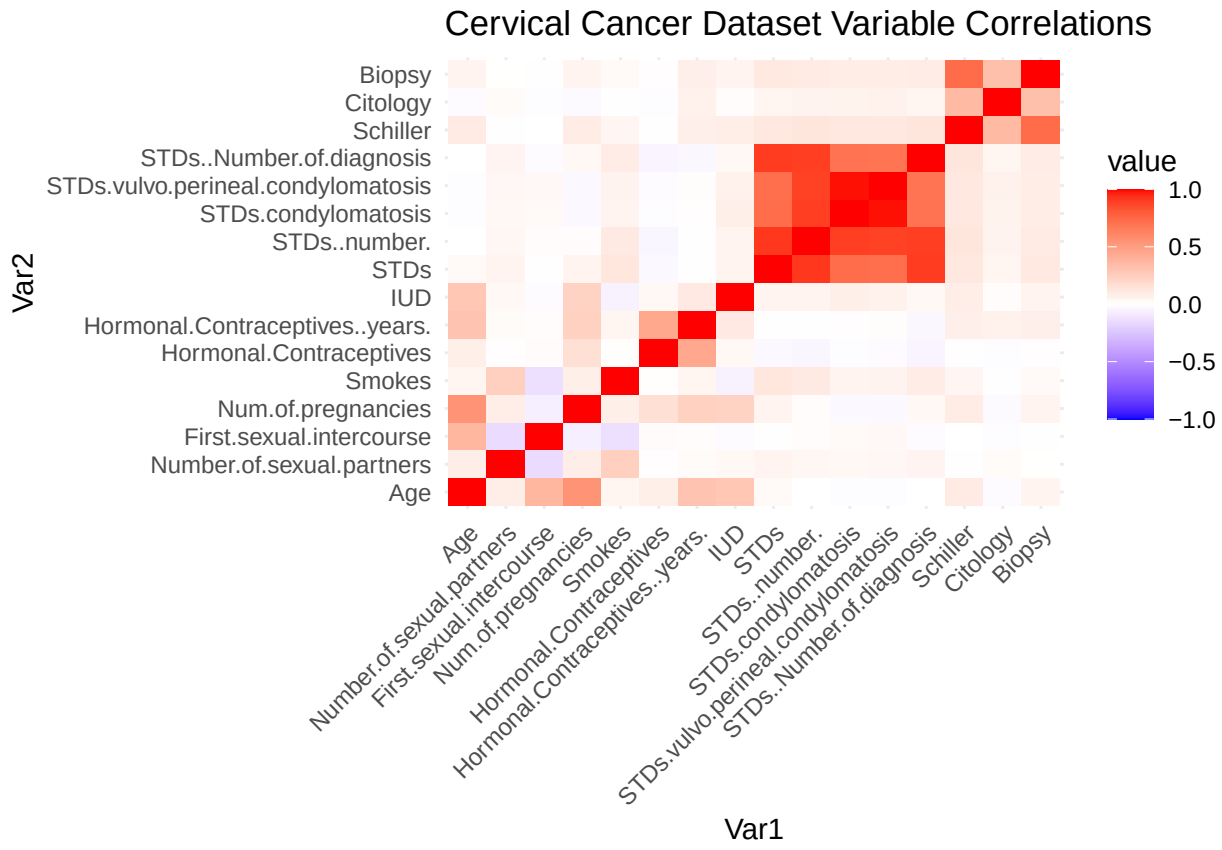
```r
# impute missing values with knn
#data_clean <- impute_with_knn(cervical_data_process, k = 29) # the rule of thumbs choosing the k is th
preproc <- preProcess(cervical_data_process, method = ("knnImpute"))
data_clean <- predict(preproc, cervical_data_process)

# since knn imputation create new columns, we will exclude the new columns from our dataset
data_clean <- subset(data_clean, select = Age:Biopsy)

null_counts_clean <- check_nulls(data_clean)
```

```
##                                                                  Column
## Age                                                                 Age
## Number.of.sexual.partners                     Number.of.sexual.partners
## First.sexual.intercourse                       First.sexual.intercourse
## Num.of.pregnancies                                   Num.of.pregnancies
## Smokes                                                           Smokes
## Hormonal.Contraceptives                         Hormonal.Contraceptives
## Hormonal.Contraceptives..years.       Hormonal.Contraceptives..years.
## IUD                                                                 IUD
## STDs                                                               STDs
## STDs..number.                                             STDs..number.
## STDs.condylomatosis                                 STDs.condylomatosis
## STDs.vulvo.perineal.condylomatosis STDs.vulvo.perineal.condylomatosis
## STDs..Number.of.diagnosis                     STDs..Number.of.diagnosis
## Schiller                                                       Schiller
## Citology                                                       Citology
## Biopsy                                                           Biopsy
## 17                                                                Total
##                                        Nulls      ColumnPercentage
## Age                                        0                     0
## Number.of.sexual.partners                  0                     0
## First.sexual.intercourse                   0                     0
## Num.of.pregnancies                         0                     0
## Smokes                                     0                     0
## Hormonal.Contraceptives                    0                     0
## Hormonal.Contraceptives..years.            0                     0
## IUD                                        0                     0
## STDs                                       0                     0
## STDs..number.                              0                     0
## STDs.condylomatosis                        0                     0
## STDs.vulvo.perineal.condylomatosis         0                     0
## STDs..Number.of.diagnosis                  0                     0
## Schiller                                   0                     0
## Citology                                   0                     0
## Biopsy                                     0                     0
## 17                                total_nulls total_percentage_null
```

EDA - Correlations Analysis

```r
# convert factor to numeric
data_clean$Biopsy <- as.numeric(data_clean$Biopsy)
```

```
# Feed into our heatmap function
heatmap <- create_heatmap("Cervical Cancer Dataset Variable Correlations", data_clean)

# Display the heatmap
print(heatmap)
```




```
ggsave(filename = "cor-matrix.png", plot = heatmap, width = 7, height = 7)
```

Check highly correlated predictors

```
highlyCorrelated <- findCorrelation(cor(data_clean), cutoff = 0.9)

print(names(data_clean)[highlyCorrelated])
```

```
## [1] "STDs..number."       "STDs"                "STDs.condylomatosis"
```

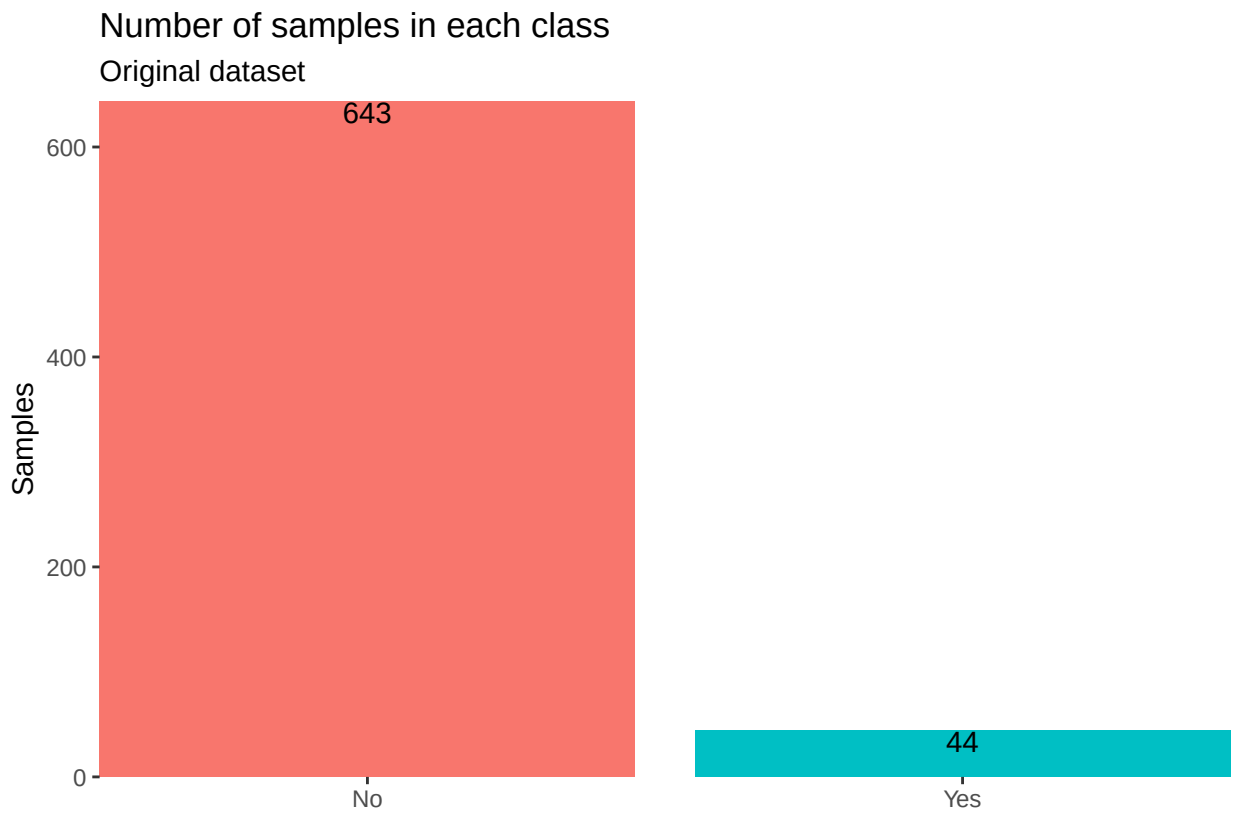
```
# drop highly correlated variables
data_clean <- data_clean[, -highlyCorrelated]
```

Convert the class to factor variable

```
# initial look at the target variable
data_clean$Biopsy<-as.factor(data_clean$Biopsy) # convert class to factor
levels(data_clean$Biopsy) <- c("No", "Yes") # names of the factors
```

Data Partitioning (Train and Test Split)

```
# data splitting
set.seed(100)

trainIndex <- createDataPartition(data_clean$Biopsy, p = .8, list = FALSE)
trainData <- data_clean[trainIndex, ]
testData  <- data_clean[-trainIndex, ]

train_X <- trainData[ , !(names(trainData) %in% "Biopsy")]
train_y <- trainData$Biopsy

test_X <- testData[ , !(names(testData) %in% "Biopsy")]
test_y <- testData$Biopsy


############################################### Imbalance class

# plotting number of samples in each class - original dataset
options(scipen=10000)

train_y_df <- data.frame(Biopsy = train_y)

# Create the plot
p <- ggplot(data = train_y_df, aes(x = Biopsy, fill = Biopsy)) +
    geom_bar() +
    geom_text(stat='count', aes(label=..count..), vjust=1) +
    ggtitle("Number of samples in each class", subtitle = "Original dataset") +
    xlab("") +
    ylab("Samples") +
    scale_y_continuous(expand = c(0,0)) +
    scale_x_discrete(expand = c(0,0)) +
    theme(legend.position = "none",
          legend.title = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.background = element_blank())
p
```

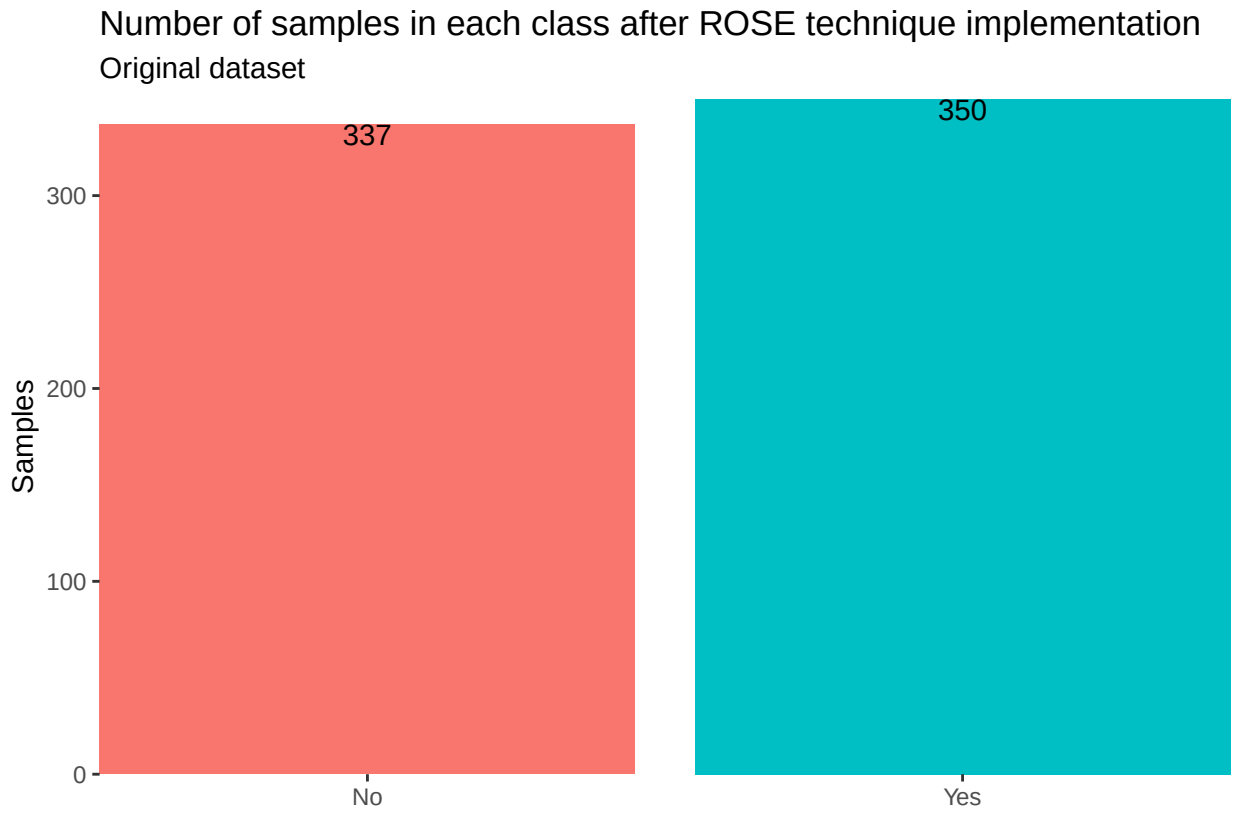



```
ggsave(filename = "class_imbalance1.png", plot = p, width = 7, height = 7)
```

Class Imbalanace (ROSE)

```
#Implementing ROSE function to handle class imbalance problem

library(ROSE)

set.seed(100)

rose_train <- ROSE(Biopsy ~ ., data = trainData)$data

train_X <- rose_train[ , !(names(rose_train) %in% "Biopsy")]
train_y <- rose_train$Biopsy


options(scipen=10000)

train_y_df <- data.frame(Biopsy = train_y)

p1 <- ggplot(data = train_y_df, aes(x = Biopsy,fill = Biopsy)) +
    geom_bar()+
    geom_text(stat='count', aes(label=..count..), vjust=1) +
    ggtitle("Number of samples in each class after ROSE technique implementation", subtitle = "Original
```

```
    xlab("")+
    ylab("Samples")+
    scale_y_continuous(expand = c(0,0))+
    scale_x_discrete(expand = c(0,0))+
    theme(legend.position = "none",
          legend.title = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.background = element_blank())
p1
```




```
ggsave(filename = "class_imbalance2.png", plot = p1, width = 7, height = 4)
```

Data Pre-Processing

```
preProcValues <- preProcess(train_X,
                            method = c("center", "scale"))

train_X <- predict(preProcValues, train_X)
test_X <- predict(preProcValues, test_X)


cntrl <- trainControl(method = "cv", number = 10,
```

```
                    summaryFunction = twoClassSummary,
                    classProbs = TRUE,
                    savePredictions = TRUE)
```

Modeling

Non-Linear models

Neural Network Model

```
### Neural Network Model
nnet_model <- train_nnet_model(train_X, train_y, ncol(trainData), cntrl)
```

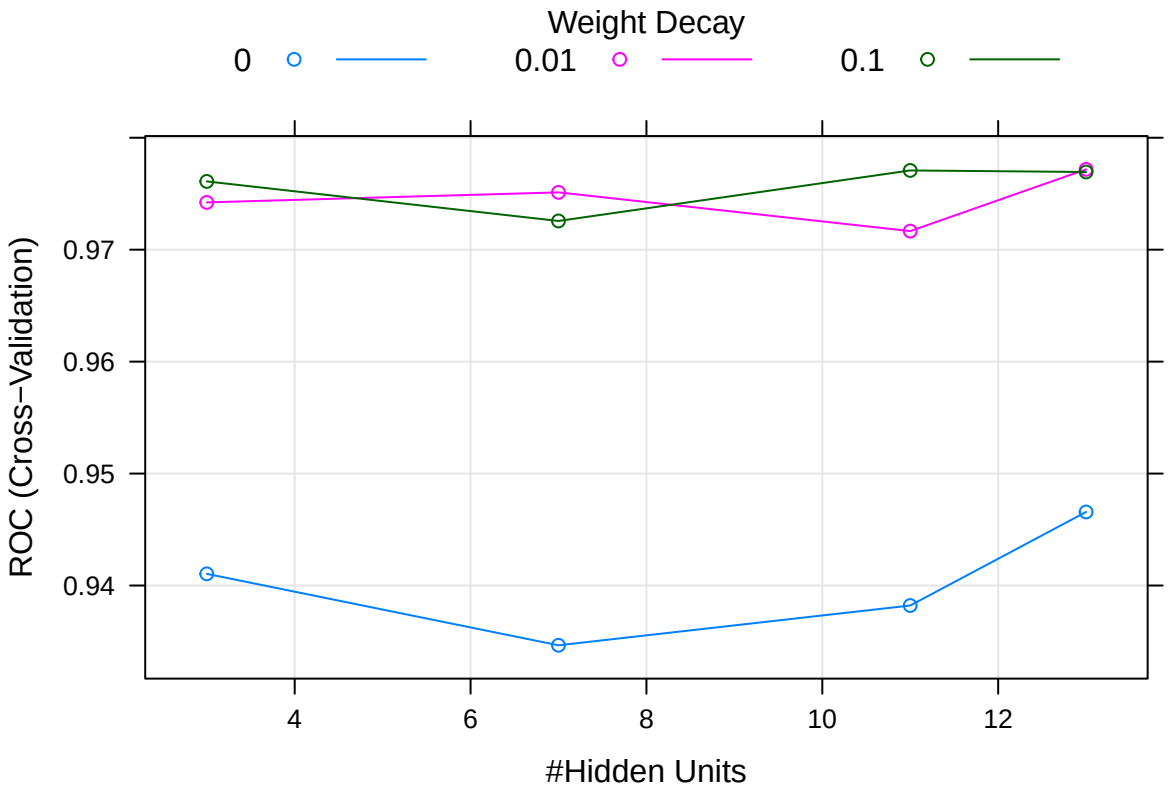
```
## Warning in train.default(x = train_X, y = train_y, method = "nnet", tuneGrid =
## nnetGrid, : The metric "Accuracy" was not in the result set. ROC will be used
## instead.
```

```
# get prediction result
testResults_nnet <- get_prediction_results(nnet_model, test_X, test_y)

# convert prediction levels to match observation
testResults_nnet$prediction <- ifelse(testResults_nnet$prediction == "1", "Yes", "No")

# confusion matrix
cm <- confusionMatrix(as.factor(testResults_nnet$prediction), as.factor(testResults_nnet$observation))
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  151   3
##        Yes   9   8
##
##                Accuracy : 0.9298
##                  95% CI : (0.8806, 0.9632)
##     No Information Rate : 0.9357
##     P-Value [Acc > NIR] : 0.6924
##
##                   Kappa : 0.5351
##
##  Mcnemar's Test P-Value : 0.1489
##
##             Sensitivity : 0.9437
##             Specificity : 0.7273
##          Pos Pred Value : 0.9805
##          Neg Pred Value : 0.4706
##              Prevalence : 0.9357
##          Detection Rate : 0.8830
```

```
##    Detection Prevalence : 0.9006
##       Balanced Accuracy : 0.8355
##
##        'Positive' Class : No
##
```

```
# neural network model result plot
plot(nnet_model)
```




```
nnet_model$finalModel
```

```
## a 12-13-1 network with 183 weights
## inputs: Age Number.of.sexual.partners First.sexual.intercourse Num.of.pregnancies Smokes Hormonal.Co
## output(s): .outcome
## options were - entropy fitting  decay=0.01
```

```
# roc/auc result
roc_nnet <- roc(testResults_nnet$observation, testResults_nnet$class_prob)
```

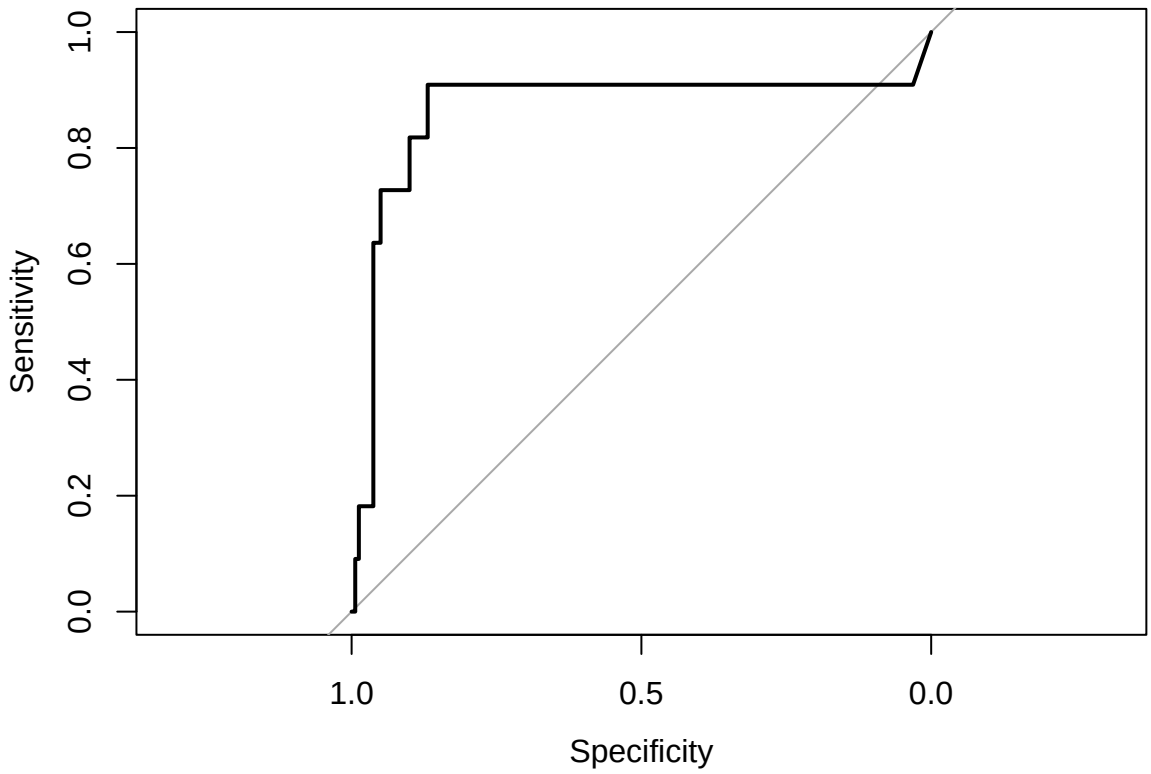
```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
auc(roc_nnet)
```

```
## Area under the curve: 0.8662
```

```
plot(roc_nnet)
```



Multivariate Adaptive Regression Splines (MARS)

```
mars_model <- train_mars_model(train_X, train_y, 2:20, cntrl)
```

```
## Warning in train.default(x = train_X, y = train_y, method = "earth", tuneGrid =
## expand.grid(degree = 1, : The metric "Accuracy" was not in the result set. ROC
## will be used instead.
```

```
## Loading required package: earth
```

```
## Loading required package: Formula
```

```
## Loading required package: plotmo
```

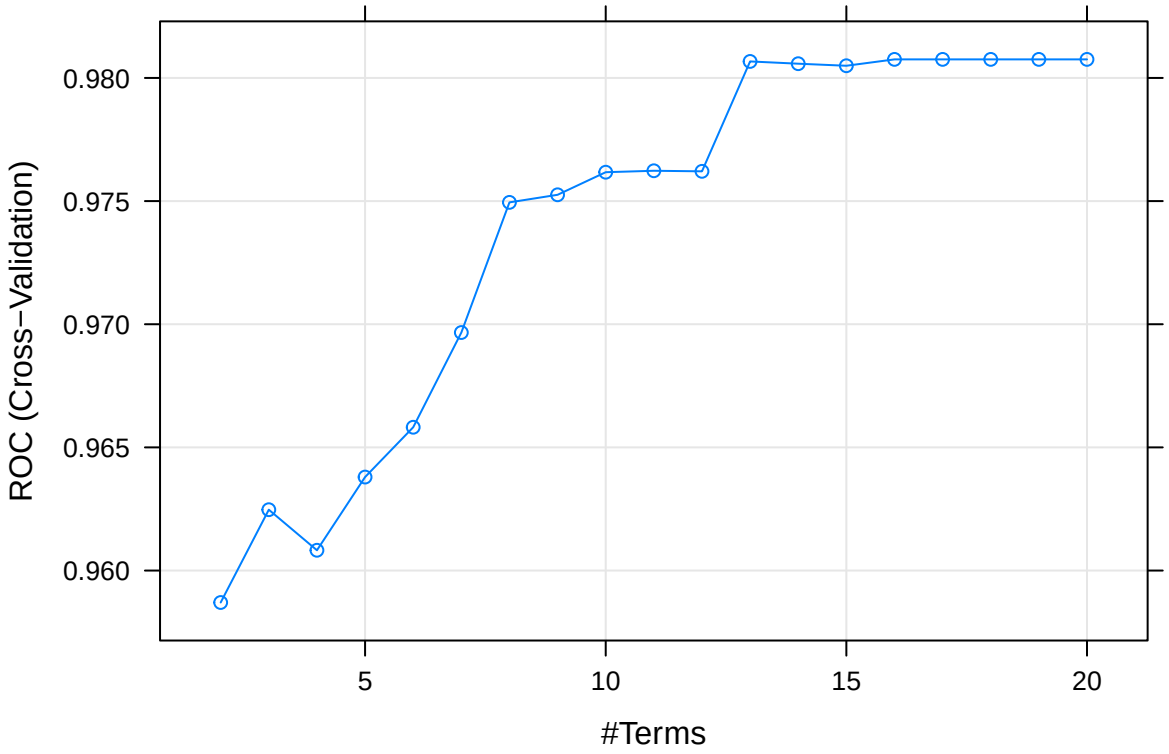
```
## Loading required package: plotrix
```

```
## Loading required package: TeachingDemos

##
## Attaching package: 'TeachingDemos'

## The following objects are masked from 'package:Hmisc':
##
##     cnvrt.coords, subplot

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
# get prediction result
testResults_mars <- get_prediction_results(mars_model, test_X, test_y)

# convert prediction levels to match observation
testResults_mars$prediction <- ifelse(testResults_mars$prediction == "1", "Yes", "No")

# confusion matrix
cm <- confusionMatrix(as.factor(testResults_mars$prediction), as.factor(testResults_mars$observation))
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  151   3
##        Yes   9   8
##
##                Accuracy : 0.9298
##                  95% CI : (0.8806, 0.9632)
##     No Information Rate : 0.9357
##     P-Value [Acc > NIR] : 0.6924
##
##                   Kappa : 0.5351
##
##  Mcnemar's Test P-Value : 0.1489
##
##             Sensitivity : 0.9437
##             Specificity : 0.7273
##          Pos Pred Value : 0.9805
##          Neg Pred Value : 0.4706
##              Prevalence : 0.9357
##          Detection Rate : 0.8830
##    Detection Prevalence : 0.9006
##       Balanced Accuracy : 0.8355
##
##        'Positive' Class : No
##
```

```r
# mars model result plot
plot(mars_model)
```




```
mars_model$finalModel
```

```
## GLM (family binomial, link logit):
##  nulldev  df       dev  df   devratio     AIC iters converged
##  952.138 686   118.065 673      0.876   146.1     9         1
##
## Earth selected 14 of 20 terms, and 7 of 12 predictors (nprune=16)
## Termination condition: Reached nk 25
## Importance: Schiller, STDs..Number.of.diagnosis, First.sexual.intercourse, ...
## Number of terms at each degree of interaction: 1 13 (additive model)
## Earth GCV 0.03567786    RSS 22.62195    GRSq 0.8576527    RSq 0.8682384
```

```
# roc/auc result
roc_mars <- roc(testResults_mars$observation, testResults_mars$class_prob)
```

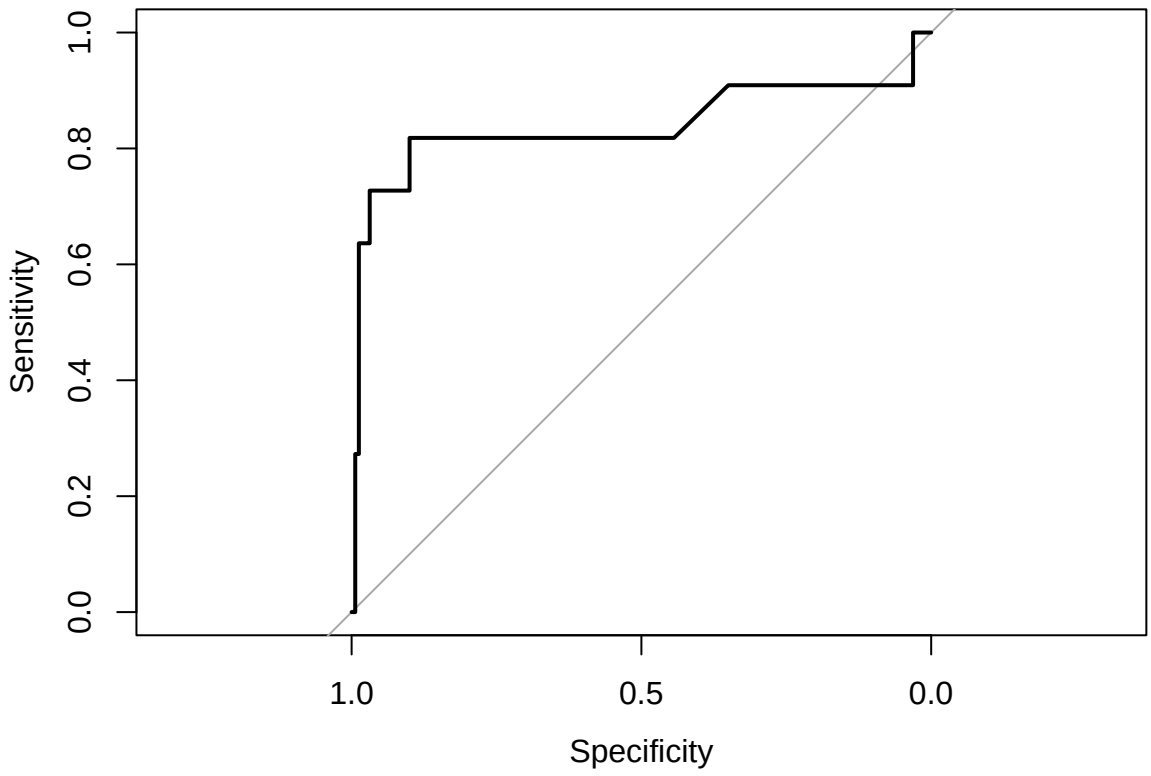
```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
auc(roc_mars)
```

```
## Area under the curve: 0.8389
```

```
plot(roc_mars)
```



Support Vector Machine (SVM)

```
svm_model <- train_svm_model(train_X, train_y, 20, cntrl)
```

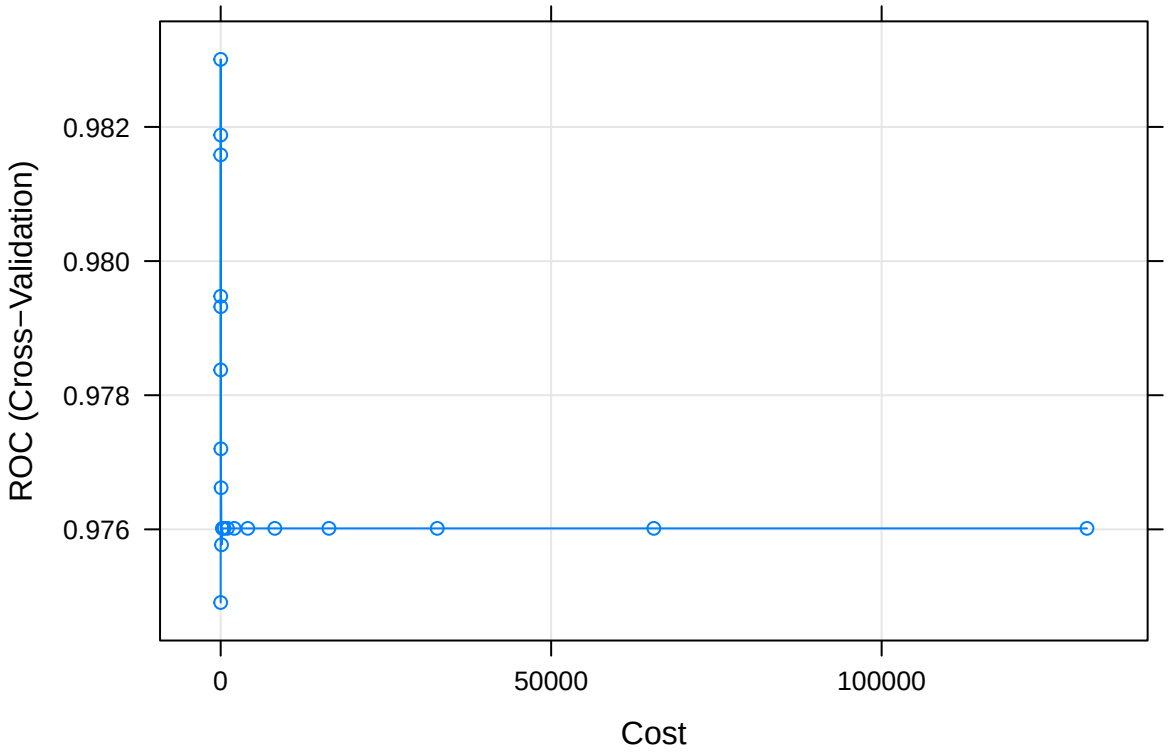
```
# get prediction result
testResults_svm <- get_prediction_results(svm_model, test_X, test_y)

# convert prediction levels to match observation
testResults_svm$prediction <- ifelse(testResults_svm$prediction == "1", "Yes", "No")

# confusion matrix
cm <- confusionMatrix(as.factor(testResults_svm$prediction), as.factor(testResults_svm$observation))
print(cm)
```

svmRadial

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction  No Yes
##        No  150   3
##        Yes  10   8
##
##                Accuracy : 0.924
##                  95% CI : (0.8735, 0.9589)
##     No Information Rate : 0.9357
##     P-Value [Acc > NIR] : 0.78756
##
##                   Kappa : 0.5128
##
##  Mcnemar's Test P-Value : 0.09609
##
##             Sensitivity : 0.9375
##             Specificity : 0.7273
##          Pos Pred Value : 0.9804
##          Neg Pred Value : 0.4444
##              Prevalence : 0.9357
##          Detection Rate : 0.8772
##    Detection Prevalence : 0.8947
##       Balanced Accuracy : 0.8324
##
##        'Positive' Class : No
##
```

```
# svm Radial result plot
plot(svm_model)
```




```
svm_model$finalModel
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 4
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.0579991726665963
##
## Number of Support Vectors : 160
##
## Objective Function Value : -249.4782
## Training error : 0.024745
## Probability model included.
```

```
# roc/auc result
roc_svm <- roc(testResults_svm$observation, testResults_svm$class_prob)
```

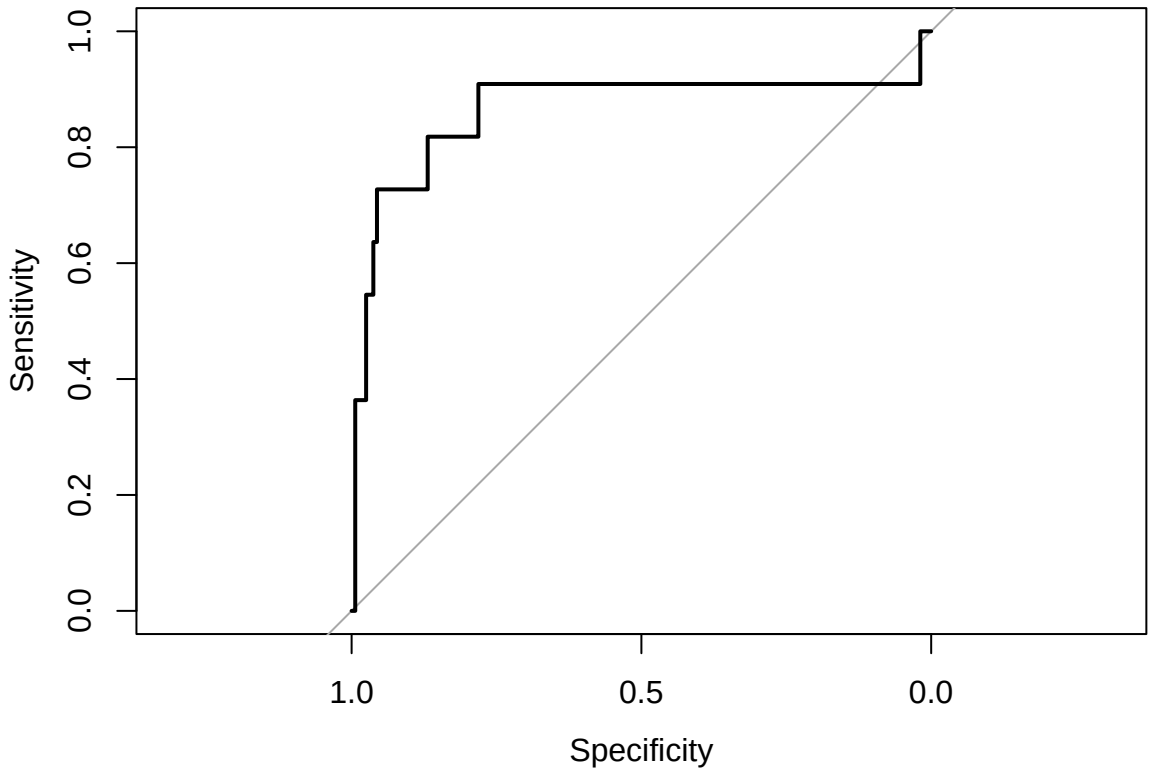
```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
auc(roc_svm)
```

```
## Area under the curve: 0.8648
```

```
plot(roc_svm)
```



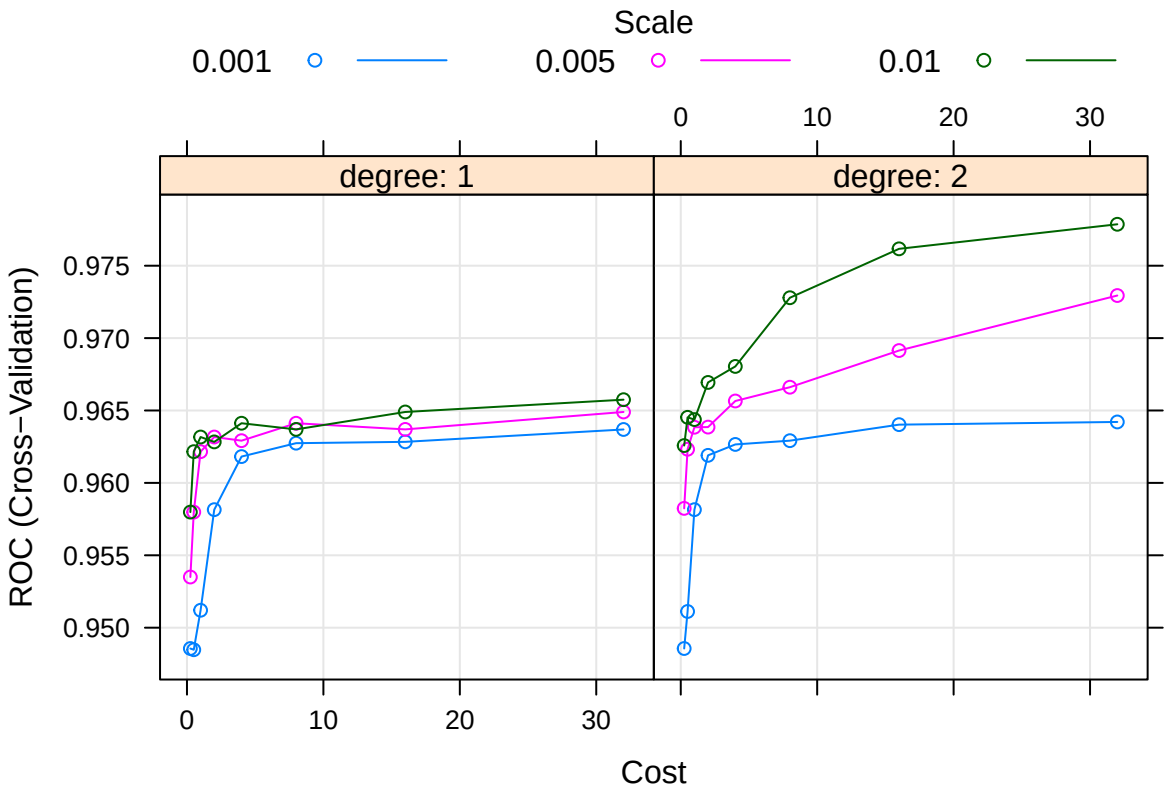

```
svm_modelPoly <- train_svm_poly(train_X, train_y, cntrl)
```

```
# get prediction result
testResults_svmP <- get_prediction_results(svm_modelPoly, test_X, test_y)

# convert prediction levels to match observation
testResults_svmP$prediction <- ifelse(testResults_svmP$prediction == "1", "Yes", "No")

# confusion matrix
cm <- confusionMatrix(as.factor(testResults_svmP$prediction), as.factor(testResults_svmP$observation))
print(cm)
```

svmPoly

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  152   3
##        Yes   8   8
##
##                Accuracy : 0.9357
##                  95% CI : (0.8878, 0.9675)
##     No Information Rate : 0.9357
##     P-Value [Acc > NIR] : 0.5793
##
##                   Kappa : 0.559
##
##  Mcnemar's Test P-Value : 0.2278
##
##             Sensitivity : 0.9500
##             Specificity : 0.7273
##          Pos Pred Value : 0.9806
##          Neg Pred Value : 0.5000
##              Prevalence : 0.9357
##          Detection Rate : 0.8889
##    Detection Prevalence : 0.9064
##       Balanced Accuracy : 0.8386
##
##        'Positive' Class : No
##
```

```
# svm Poly result plot
plot(svm_modelPoly)
```




```
svm_modelPoly$finalModel
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 32
##
## Polynomial kernel function.
##  Hyperparameters : degree =  2  scale =  0.01  offset =  1
##
## Number of Support Vectors : 107
##
## Objective Function Value : -2296.99
## Training error : 0.03639
## Probability model included.
```

```
# roc/auc result
roc_svmp <- roc(testResults_svmP$observation, testResults_svmP$class_prob)
```

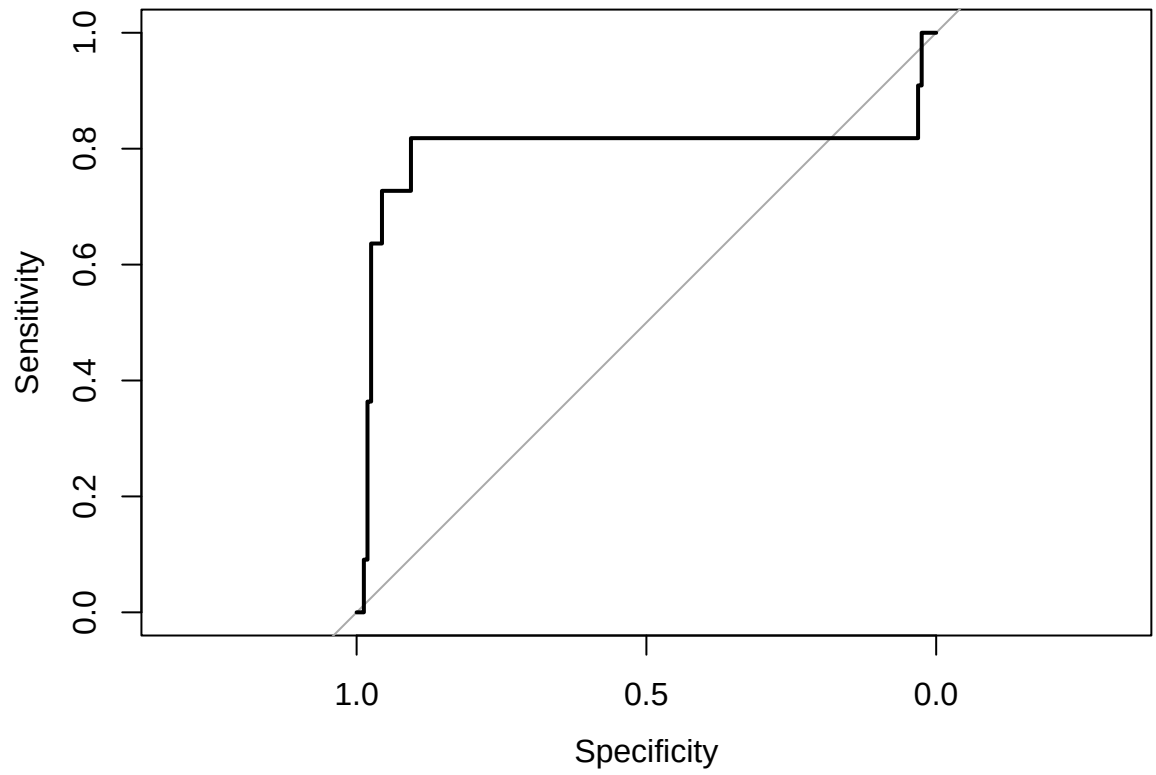
```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
auc(roc_svmp)
```

```
## Area under the curve: 0.7977
```

```
plot(roc_svmp)
```



K-Nearest Neighbors

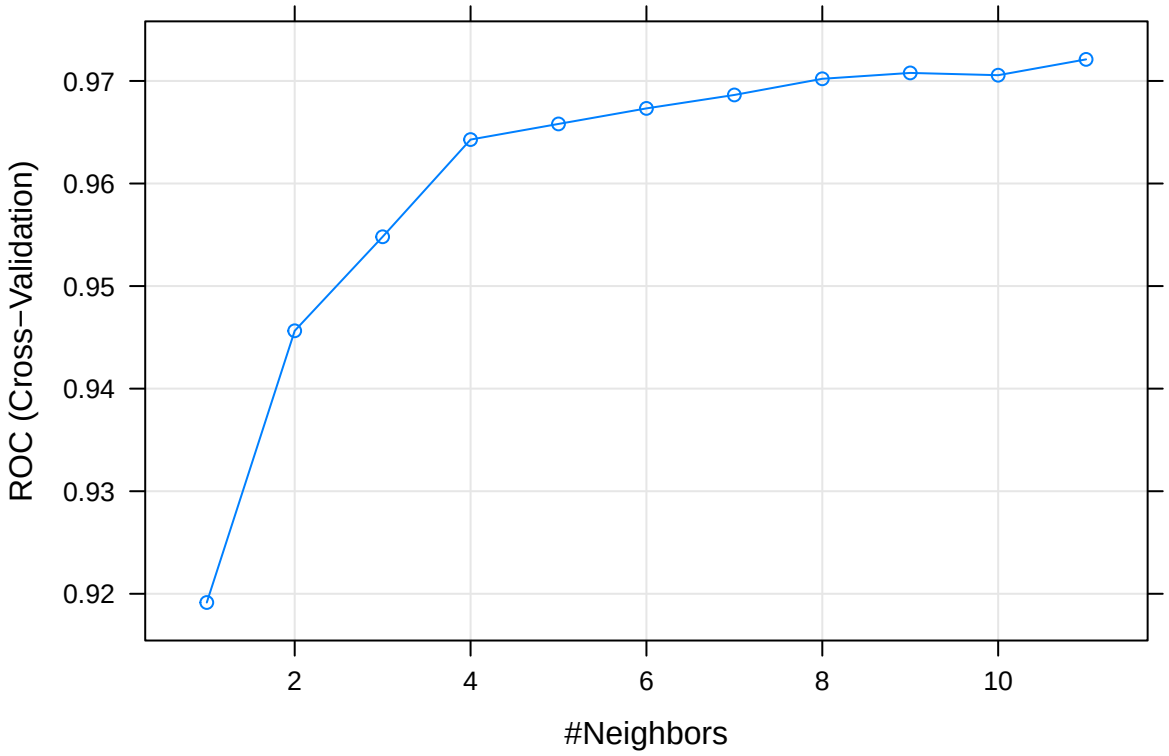
```
knn_model <- knn_model_train(train_X, train_y, cntrl, 1:11)
```

```
## Warning in train.default(x = train_X, y = train_y, method = "knn", tuneGrid =
## knnGrid, : The metric "Accuracy" was not in the result set. ROC will be used
## instead.
```

```
# get prediction result
testResults_knn <- get_prediction_results(knn_model, test_X, test_y)


# convert prediction levels to match observation
testResults_knn$prediction <- ifelse(testResults_knn$prediction == "1", "Yes", "No")

# confusion matrix
cm <- confusionMatrix(as.factor(testResults_knn$prediction), as.factor(testResults_knn$observation))
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  150   3
##        Yes  10   8
##
##                Accuracy : 0.924
##                  95% CI : (0.8735, 0.9589)
##     No Information Rate : 0.9357
##     P-Value [Acc > NIR] : 0.78756
##
##                   Kappa : 0.5128
##
##  Mcnemar's Test P-Value : 0.09609
##
##             Sensitivity : 0.9375
##             Specificity : 0.7273
##          Pos Pred Value : 0.9804
##          Neg Pred Value : 0.4444
##              Prevalence : 0.9357
##          Detection Rate : 0.8772
##    Detection Prevalence : 0.8947
##       Balanced Accuracy : 0.8324
##
##        'Positive' Class : No
##
```

```
# kNN result plot
plot(knn_model)
```




```
knn_model$finalModel
```

```
## 11-nearest neighbor model
## Training set outcome distribution:
##
##  No Yes
## 337 350
```

```
# roc/auc result
roc_knn <- roc(testResults_knn$observation, testResults_knn$class_prob)
```

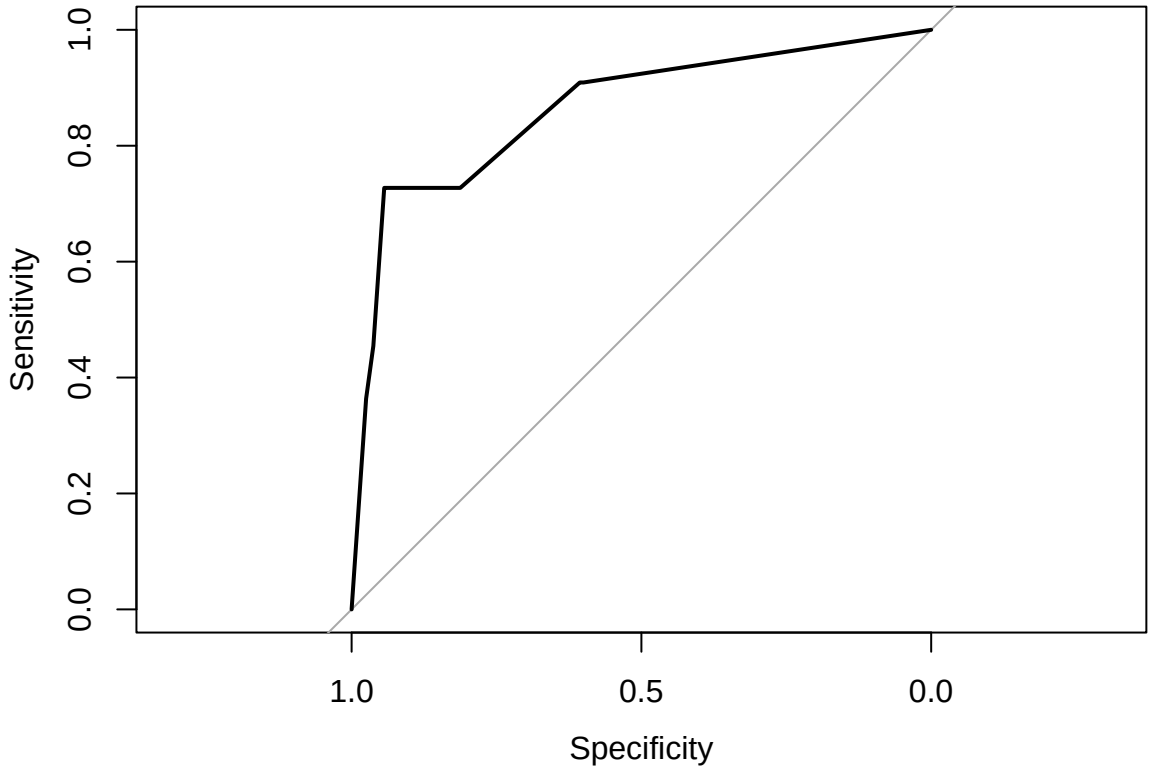
```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
auc(roc_knn)
```

```
## Area under the curve: 0.8634
```

```
plot(roc_knn)
```



Random Forest Model

```
rf_model <- rf_model_train(train_X, train_y, cntrl)
```

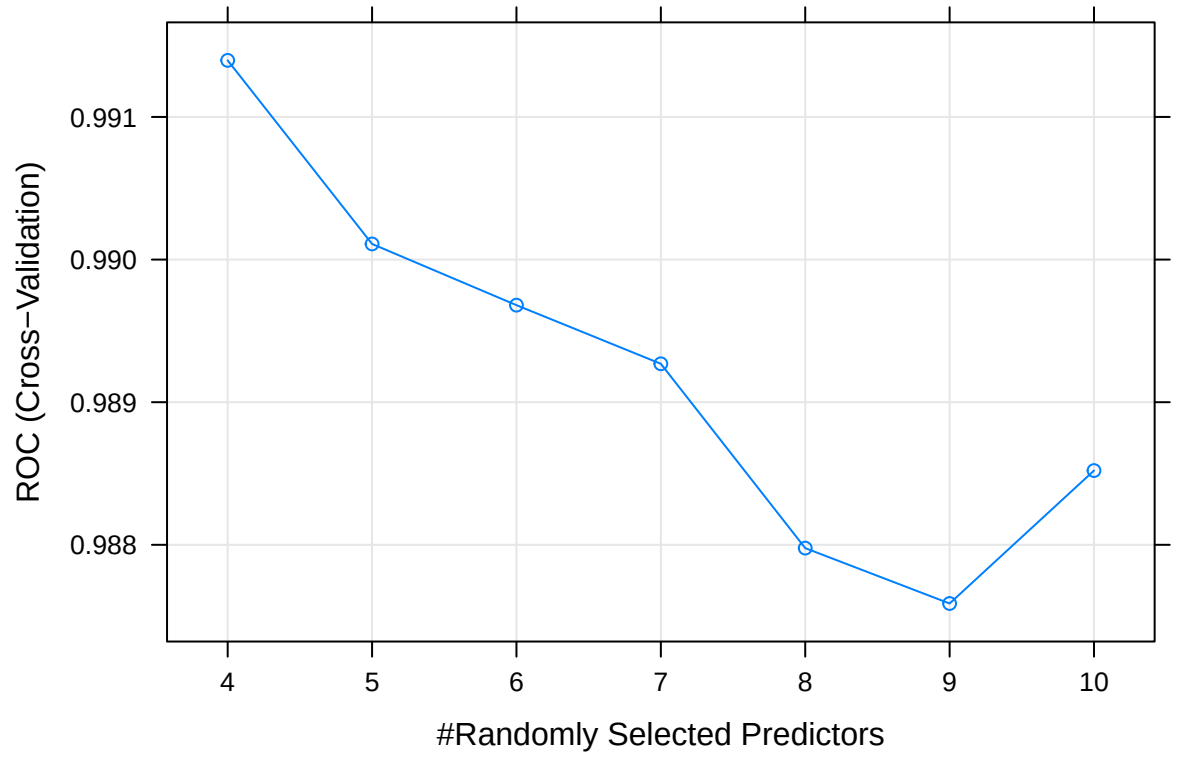
```
## Warning in train.default(x = train_X, y = train_y, method = "rf", tuneGrid =
## mtryGrid, : The metric "Accuracy" was not in the result set. ROC will be used
## instead.
```

```
# get prediction result
testResults_rf <- get_prediction_results(rf_model, test_X, test_y)

# convert prediction levels to match observation
testResults_rf$prediction <- ifelse(testResults_rf$prediction == "1", "Yes", "No")

# confusion matrix
cm <- confusionMatrix(as.factor(testResults_rf$prediction), as.factor(testResults_rf$observation))
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  152   3
##        Yes   8   8
##
##                Accuracy : 0.9357
```

```
##                  95% CI : (0.8878, 0.9675)
##     No Information Rate : 0.9357
##     P-Value [Acc > NIR] : 0.5793
##
##                   Kappa : 0.559
##
##  Mcnemar's Test P-Value : 0.2278
##
##             Sensitivity : 0.9500
##             Specificity : 0.7273
##          Pos Pred Value : 0.9806
##          Neg Pred Value : 0.5000
##              Prevalence : 0.9357
##          Detection Rate : 0.8889
##    Detection Prevalence : 0.9064
##       Balanced Accuracy : 0.8386
##
##        'Positive' Class : No
##
```

```
# RF result plot
plot(rf_model)
```




```
rf_model$finalModel
```

```
##
```

```
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 4.95%
## Confusion matrix:
##      No Yes class.error
## No  327  10  0.02967359
## Yes  24 326  0.06857143
```

```
# roc/auc result
roc_rf <- roc(testResults_rf$observation, testResults_rf$class_prob)
```

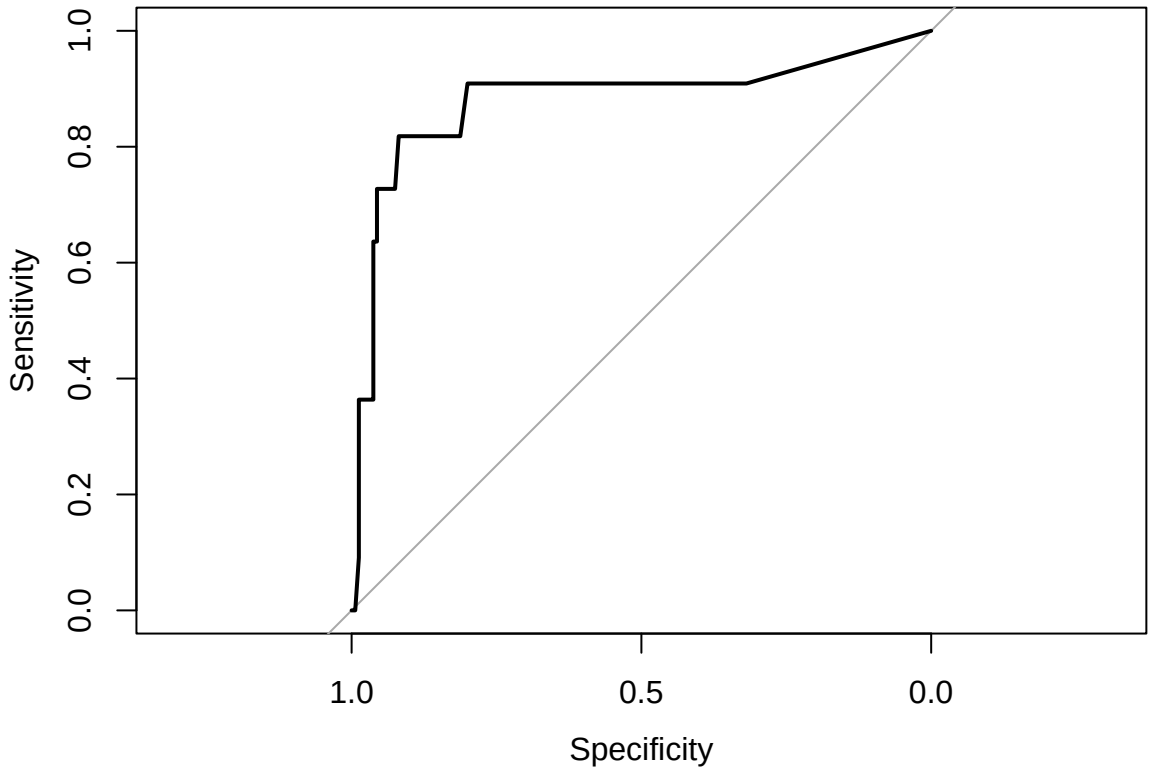
```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
auc(roc_rf)
```

```
## Area under the curve: 0.8804
```

```
plot(roc_rf)
```



Linear Model

Logistic Regression

```
lr_model <- lr_model_train(train_X, train_y, cntrl)
```

```
# get prediction result
testResults_lr <- get_prediction_results(lr_model, test_X, test_y)


# convert prediction levels to match observation
testResults_lr$prediction <- ifelse(testResults_lr$prediction == "1", "Yes", "No")


# confusion matrix
cm <- confusionMatrix(as.factor(testResults_lr$prediction), as.factor(testResults_lr$observation))
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  151   3
##        Yes   9   8
##
##                Accuracy : 0.9298
##                  95% CI : (0.8806, 0.9632)
##     No Information Rate : 0.9357
##     P-Value [Acc > NIR] : 0.6924
##
##                   Kappa : 0.5351
##
##  Mcnemar's Test P-Value : 0.1489
##
##             Sensitivity : 0.9437
##             Specificity : 0.7273
##          Pos Pred Value : 0.9805
##          Neg Pred Value : 0.4706
##              Prevalence : 0.9357
##          Detection Rate : 0.8830
##    Detection Prevalence : 0.9006
##       Balanced Accuracy : 0.8355
##
##        'Positive' Class : No
##
```

```
# roc/auc result

roc_lr <- roc(testResults_lr$observation, testResults_lr$class_prob)
```

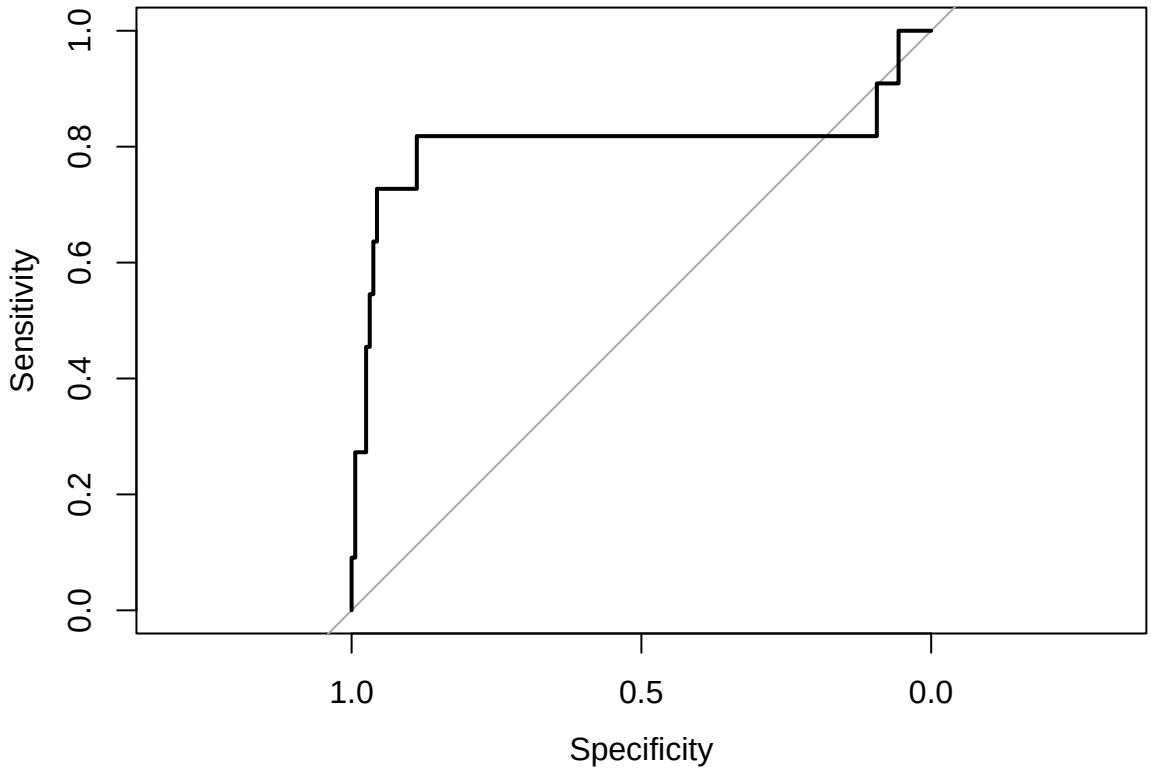
```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
auc(roc_lr)
```

```
## Area under the curve: 0.8057
```

```
plot(roc_lr)
```



LDA Model

```
lda_model <- lda_model_train(train_X, train_y, cntrl)
```

```
# get prediction result
testResults_lda <- get_prediction_results(lda_model, test_X, test_y)

# convert prediction levels to match observation
testResults_lda$prediction <- ifelse(testResults_lda$prediction == "1", "Yes", "No")


# confusion matrix
cm <- confusionMatrix(as.factor(testResults_lda$prediction), as.factor(testResults_lda$observation))
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  152   3
##        Yes   8   8
##
##                Accuracy : 0.9357
##                  95% CI : (0.8878, 0.9675)
##     No Information Rate : 0.9357
##     P-Value [Acc > NIR] : 0.5793
##
##                   Kappa : 0.559
##
##  Mcnemar's Test P-Value : 0.2278
##
##             Sensitivity : 0.9500
##             Specificity : 0.7273
##          Pos Pred Value : 0.9806
##          Neg Pred Value : 0.5000
##              Prevalence : 0.9357
##          Detection Rate : 0.8889
##    Detection Prevalence : 0.9064
##       Balanced Accuracy : 0.8386
##
##        'Positive' Class : No
##
```

```
# roc/auc result
roc_lda <- roc(testResults_lda$observation, testResults_lda$class_prob)
```

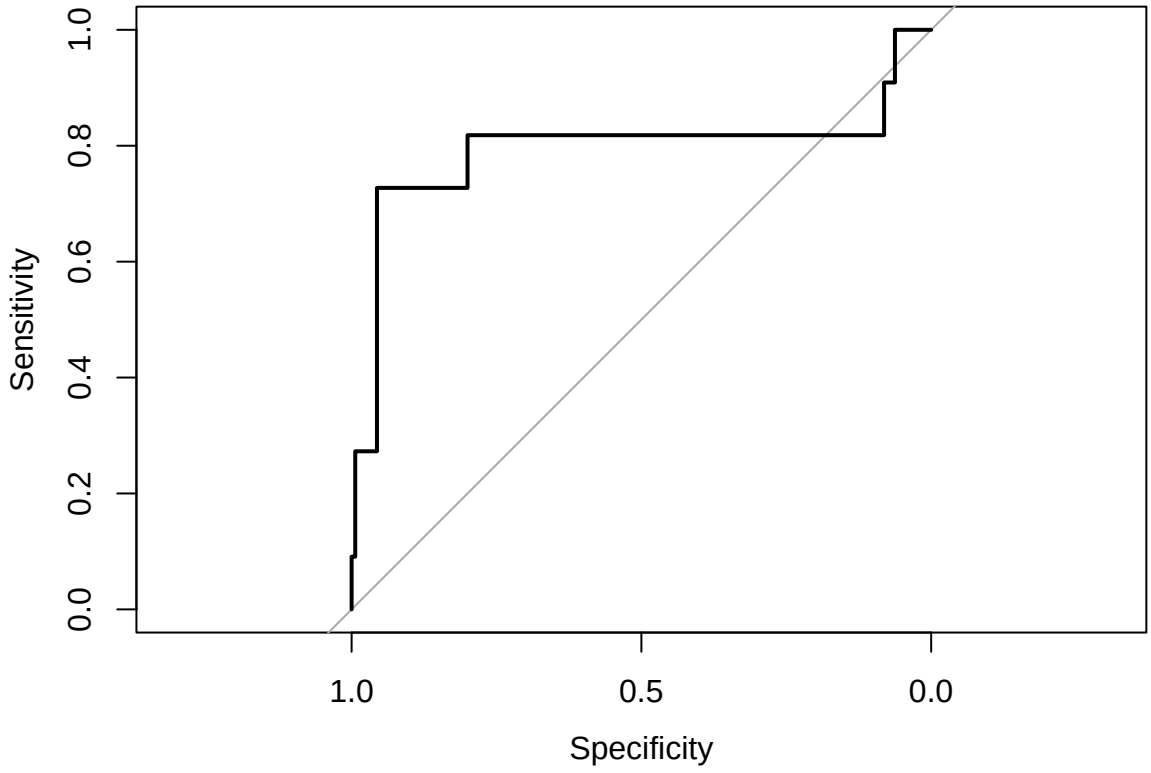
```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
auc(roc_lda)
```

```
## Area under the curve: 0.792
```

```
plot(roc_lda)
```



Penalized Logistic Regression

```
glmn_model <- glmn_model_train(train_X, train_y, cntrl)
```

```
# get prediction result
testResults_glmn <- get_prediction_results(glmn_model, test_X, test_y)

# convert prediction levels to match observation
testResults_glmn$prediction <- ifelse(testResults_glmn$prediction == "1", "Yes", "No")

# confusion matrix
cm <- confusionMatrix(as.factor(testResults_glmn$prediction), as.factor(testResults_glmn$observation))
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  152   3
##        Yes   8   8
##
##                Accuracy : 0.9357
##                  95% CI : (0.8878, 0.9675)
```

```
##     No Information Rate : 0.9357
##     P-Value [Acc > NIR] : 0.5793
##
##                   Kappa : 0.559
##
##  Mcnemar's Test P-Value : 0.2278
##
##             Sensitivity : 0.9500
##             Specificity : 0.7273
##          Pos Pred Value : 0.9806
##          Neg Pred Value : 0.5000
##              Prevalence : 0.9357
##          Detection Rate : 0.8889
##    Detection Prevalence : 0.9064
##       Balanced Accuracy : 0.8386
##
##        'Positive' Class : No
##
```

```
# roc/auc result
roc_glmn <- roc(testResults_glmn$observation, testResults_glmn$class_prob)
```

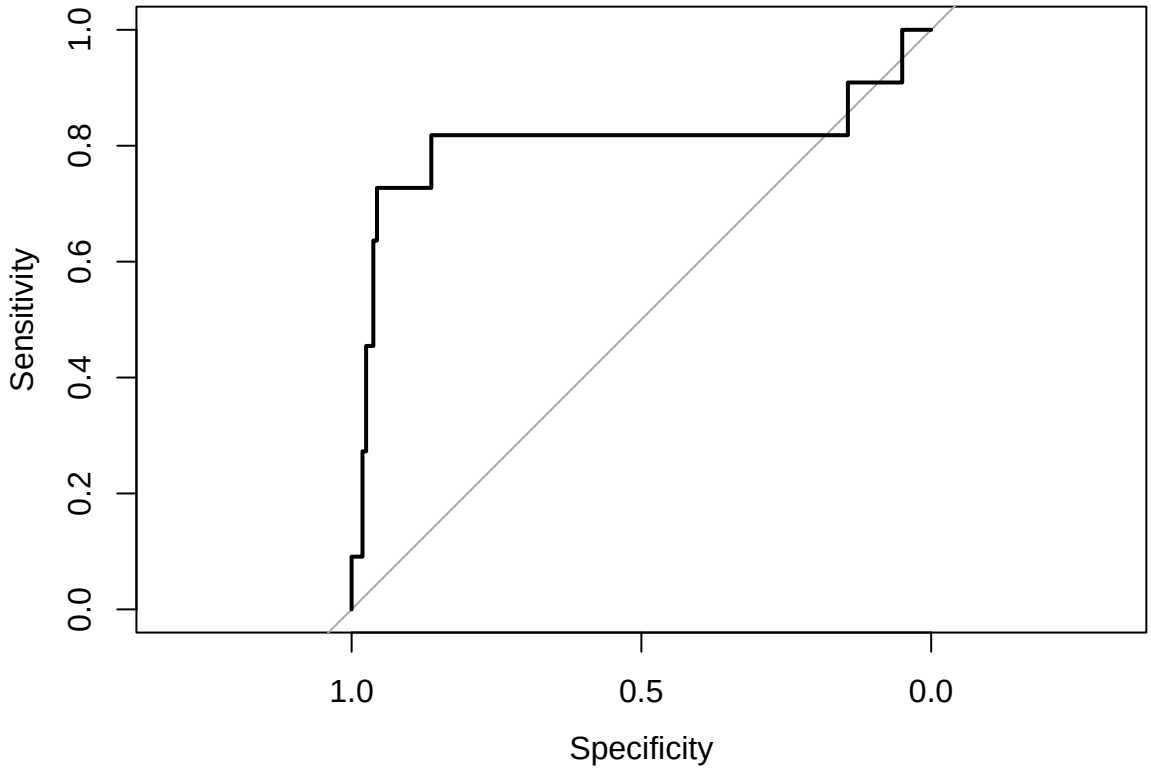
```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
auc(roc_glmn)
```

```
## Area under the curve: 0.8045
```

```
plot(roc_glmn)
```



Nearest Shrunken Centroids

```
nsc_model <- nsc_model_train(train_X, train_y, cntrl)
```

```
## 11111111111
```

```
# get prediction result
testResults_nsc <- get_prediction_results(nsc_model, test_X, test_y)

# convert prediction levels to match observation
testResults_nsc$prediction <- ifelse(testResults_nsc$prediction == "1", "Yes", "No")


# confusion matrix
cm <- confusionMatrix(as.factor(testResults_nsc$prediction), as.factor(testResults_nsc$observation))
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  152   3
##        Yes   8   8
```

```
##
##                Accuracy : 0.9357
##                  95% CI : (0.8878, 0.9675)
##     No Information Rate : 0.9357
##     P-Value [Acc > NIR] : 0.5793
##
##                   Kappa : 0.559
##
##  Mcnemar's Test P-Value : 0.2278
##
##             Sensitivity : 0.9500
##             Specificity : 0.7273
##          Pos Pred Value : 0.9806
##          Neg Pred Value : 0.5000
##              Prevalence : 0.9357
##          Detection Rate : 0.8889
##    Detection Prevalence : 0.9064
##       Balanced Accuracy : 0.8386
##
##        'Positive' Class : No
##
```

```
# roc/auc result
roc_nsc <- roc(testResults_nsc$observation, testResults_nsc$class_prob)
```

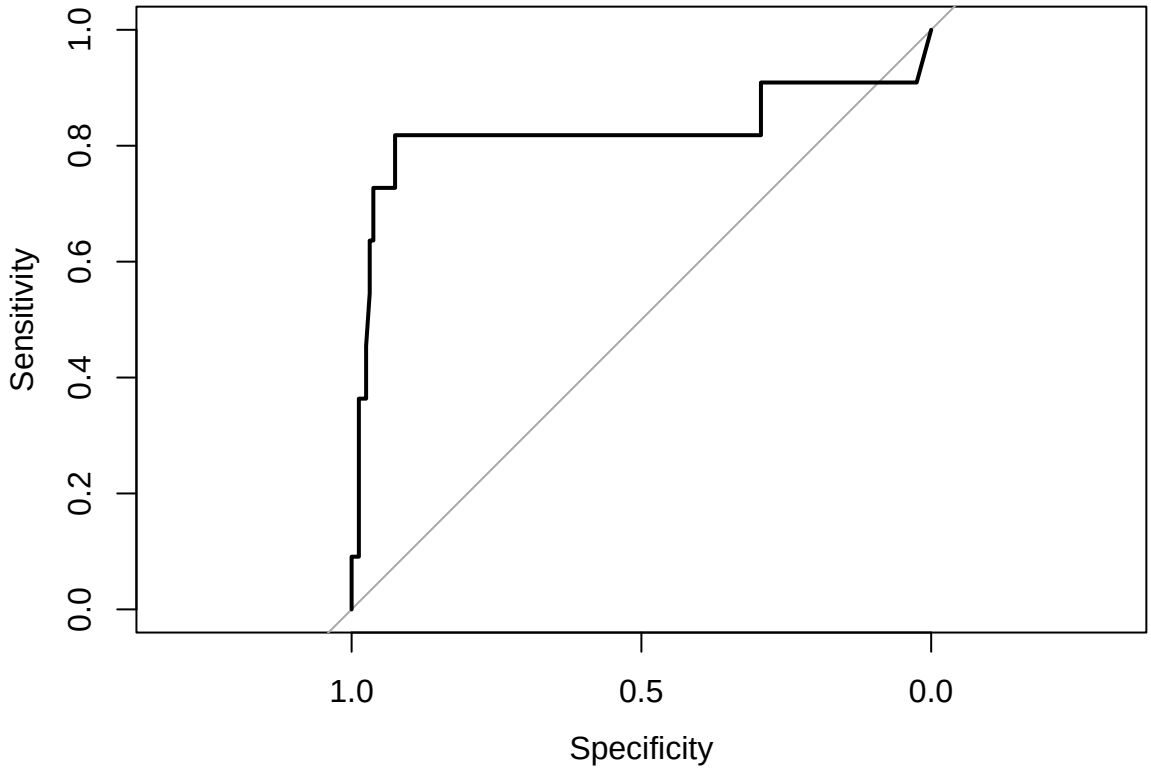
```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
auc(roc_nsc)
```

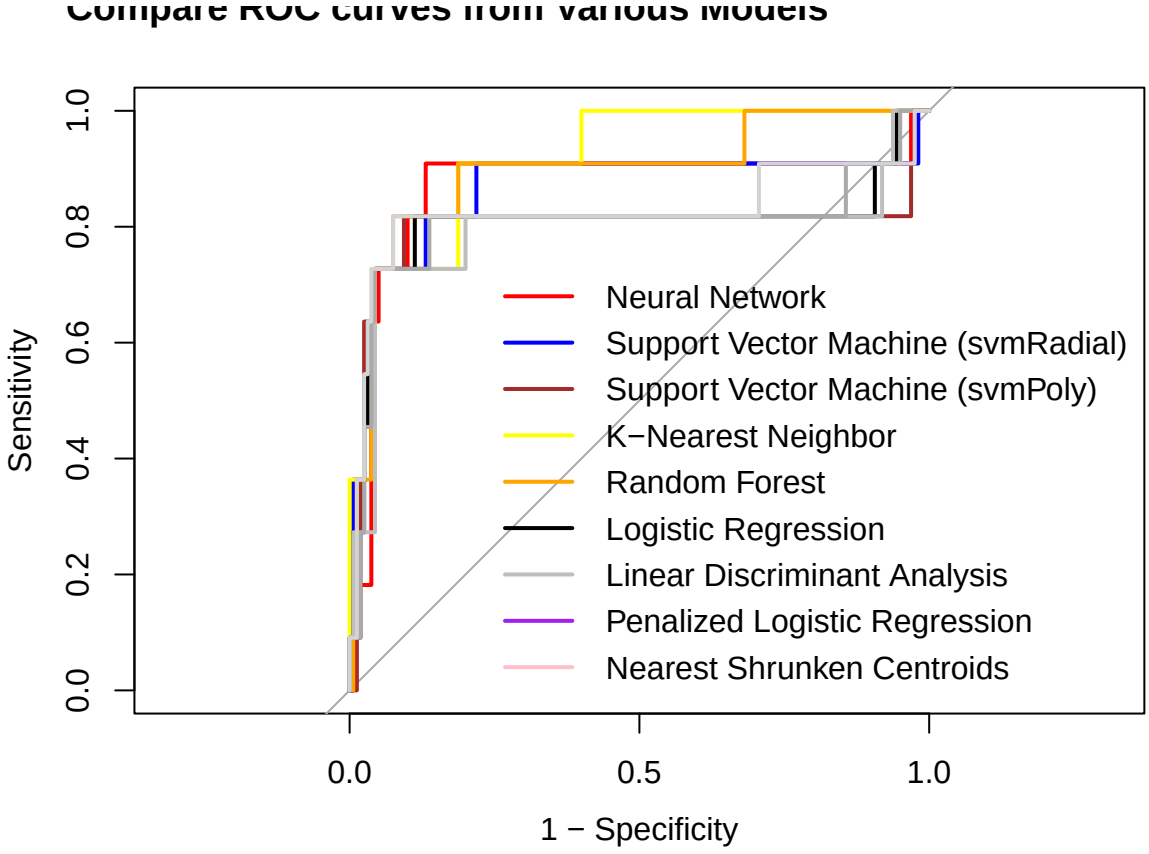
```
## Area under the curve: 0.8247
```

```
plot(roc_nsc)
```



Final Model Evaluation & Enhancements

```
### Compare Models using ROC curve
par(mar = c(9, 1, 0, 9))

# Non-linear model plots
plot(roc_nnet, type = "s", col = 'red', legacy.axes = TRUE)
plot(roc_svm, type = "s", add = TRUE, col = 'blue', legacy.axes = TRUE)
plot(roc_svmp, type = "s", add = TRUE, col = 'brown', legacy.axes = TRUE)
plot(roc_knn, type = "s", add = TRUE, col = 'yellow', legacy.axes = TRUE)
plot(roc_rf, type = "s", add = TRUE, col = 'orange', legacy.axes = TRUE)

# Linear model plots
plot(roc_lr, type = "s", add = TRUE, col = 'black', legacy.axes = TRUE)
plot(roc_lda, type = "s", add = TRUE, col = 'gray', legacy.axes = TRUE)
plot(roc_glmn, type = "s", add = TRUE, col = 'darkgray', legacy.axes = TRUE)
plot(roc_nsc, type = "s", add = TRUE, col = 'lightgray', legacy.axes = TRUE)

# Update the legend to include the new models
legend("bottomright", legend=c("Neural Network", "Support Vector Machine (svmRadial)", "Support Vector
       col=c("red", "blue", "brown", "yellow", "orange", "black", "gray", "purple", "pink"), lwd=2, bty

title(main = "Compare ROC curves from Various Models")
```



Model performance based on different metrics (AUC/ROC, Accuracy)

```
# auc result
nnetAuc <- auc(roc_nnet)
marsAuc <- auc(roc_mars)
svmAuc <- auc(roc_svm)
svmpAuc <- auc(roc_svmp)
knnAuc <- auc(roc_knn)
rfAuc <- auc(roc_rf)
lrAuc <- auc(roc_lr)
ldaAuc <- auc(roc_lda)
glmnAuc <- auc(roc_glmn)
nscAuc <- auc(roc_nsc)

# accuracy result
nnetAcc <- get_accuracy(nnet_model, test_X, test_y)
marsAcc <- get_accuracy(mars_model, test_X, test_y)
svmAcc <- get_accuracy(svm_model, test_X, test_y)
svmpAcc <- get_accuracy(svm_modelPoly, test_X, test_y)
knnAcc <- get_accuracy(knn_model, test_X, test_y)
rfAcc <- get_accuracy(rf_model, test_X, test_y)
lrAcc <- get_accuracy(lr_model, test_X, test_y)
ldaAcc <- get_accuracy(lda_model, test_X, test_y)
glmnAcc <- get_accuracy(glmn_model, test_X, test_y)
```

```r
nscAcc <- get_accuracy(nsc_model, test_X, test_y)

auc_df <- data.frame(
  Model = c("Neural Network", "MARS", "Support Vector Machine (svmRadial)", "Support Vector Machine (sv
                                "K-Nearest Neighbor", "Random Forest", "Logistic Regression", "Linear Di
                                 "Penalized Logistic Regression", "Nearest Shrunken Centroids"),

  AUC = c(nnetAuc, marsAuc, svmAuc, svmpAuc, knnAuc, rfAuc, lrAuc, ldaAuc, glmnAuc, nscAuc),
  Accuracy = c(nnetAcc, marsAcc, svmAcc, svmpAcc, knnAcc, rfAcc, lrAcc, ldaAcc, glmnAcc, nscAcc)
)

print(auc_df)
```

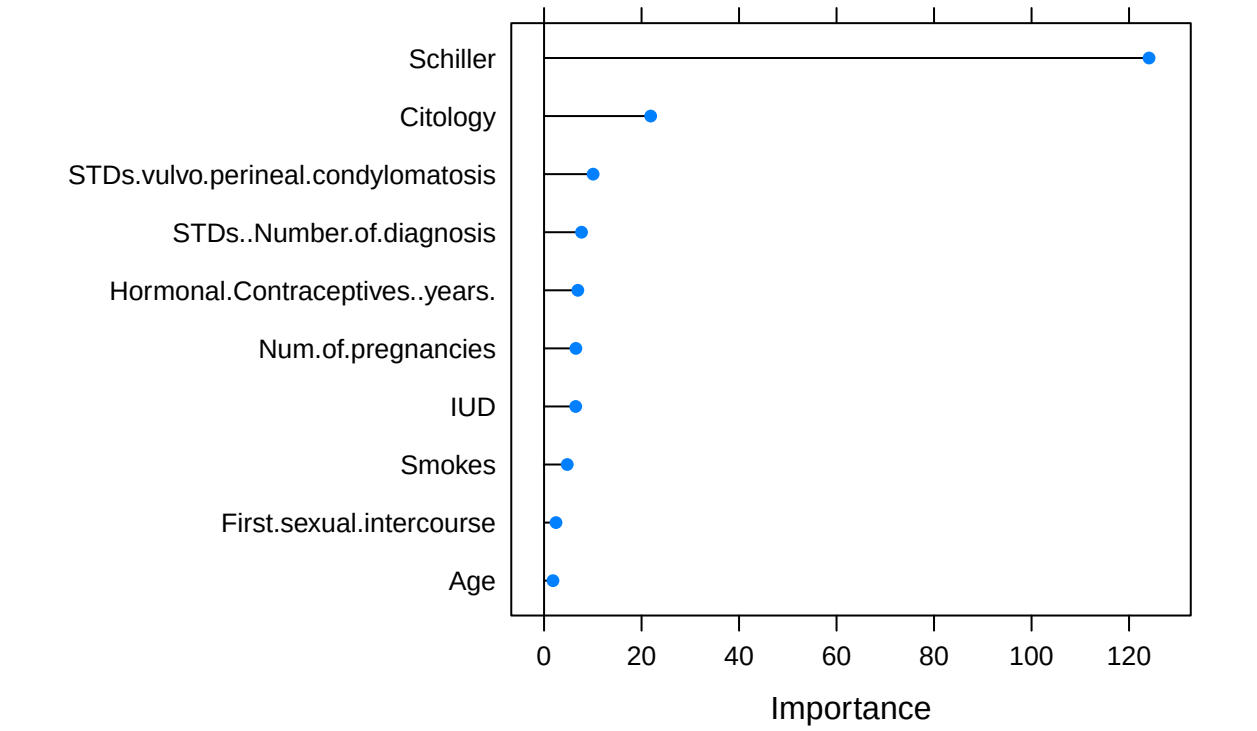
```
##                                Model       AUC  Accuracy
## 1                     Neural Network 0.8661932 0.9298246
## 2                               MARS 0.8389205 0.9298246
## 3  Support Vector Machine (svmRadial) 0.8647727 0.9239766
## 4    Support Vector Machine (svmPoly) 0.7977273 0.9356725
## 5                  K-Nearest Neighbor 0.8633523 0.9239766
## 6                       Random Forest 0.8803977 0.9356725
## 7                 Logistic Regression 0.8056818 0.9298246
## 8        Linear Discriminant Analysis 0.7920455 0.9356725
## 9       Penalized Logistic Regression 0.8045455 0.9356725
## 10         Nearest Shrunken Centroids 0.8247159 0.9356725
```

```r
# best model based on the AUC curve
best_model <- auc_df[which.max(auc_df$AUC), ]

print(best_model)
```

```
##           Model       AUC  Accuracy
## 6 Random Forest 0.8803977 0.9356725
```

Checking the important variables of the optimal model

```r
plot(varImp(rf_model, scale = FALSE), top = 10,
     main = "Important Factors for Predicting Cervical Cancer using Random Forest")
```



Recursive Feature Elimination (RFE)

```
# use caret package & user-defined-function in Modeling.R to do recursive feature elimination
optimal_rf_features <- rf_rfe(train_X, train_y)
print(optimal_rf_features)
```

```
##  [1] "Schiller"                          "Citology"
##  [3] "STDs.vulvo.perineal.condylomatosis" "Hormonal.Contraceptives..years."
##  [5] "Num.of.pregnancies"                "STDs..Number.of.diagnosis"
##  [7] "IUD"                               "Smokes"
##  [9] "First.sexual.intercourse"          "Number.of.sexual.partners"
## [11] "Age"
```

```
# Retrain penalized LR with optimal features - 12 out of 15
train_X_rfe <- train_X[, optimal_rf_features]

rf_model_rfe <- rf_model_train(train_X_rfe, train_y, cntrl)
rf_model_rfe
```

```
## Random Forest
##
## 687 samples
##  11 predictor
```

```
##   2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 618, 618, 618, 618, 619, 619, ...
## Resampling results across tuning parameters:
##
##   mtry  ROC        Sens       Spec
##    3    0.9930125  0.9762923  0.9371429
##    4    0.9912618  0.9733512  0.9371429
##    5    0.9903196  0.9645276  0.9428571
##    6    0.9889419  0.9704100  0.9400000
##    7    0.9876802  0.9645276  0.9457143
##    8    0.9875465  0.9704100  0.9400000
##    9    0.9873848  0.9644385  0.9428571
##   10    0.9879730  0.9644385  0.9457143
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 3.
```

```
# Test new model
test_X_rfe <- test_X[, optimal_rf_features]

# get prediction result
testResults_rf_rfe <- get_prediction_results(rf_model_rfe, test_X_rfe, test_y)


testResults_rf_rfe$prediction <- ifelse(testResults_rf_rfe$prediction == "1", "Yes", "No")

# confusion matrix
cm <- confusionMatrix(as.factor(testResults_rf_rfe$prediction), as.factor(testResults_rf_rfe$observatio
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  152   3
##        Yes   8   8
##
##                Accuracy : 0.9357
##                  95% CI : (0.8878, 0.9675)
##     No Information Rate : 0.9357
##     P-Value [Acc > NIR] : 0.5793
##
##                   Kappa : 0.559
##
##  Mcnemar's Test P-Value : 0.2278
##
##             Sensitivity : 0.9500
##             Specificity : 0.7273
##          Pos Pred Value : 0.9806
##          Neg Pred Value : 0.5000
##              Prevalence : 0.9357
```

```
##          Detection Rate : 0.8889
##    Detection Prevalence : 0.9064
##       Balanced Accuracy : 0.8386
##
##        'Positive' Class : No
##
```

```
# roc/auc result
roc_rf_rfe <- roc(testResults_rf_rfe$observation, testResults_rf_rfe$class_prob)
```

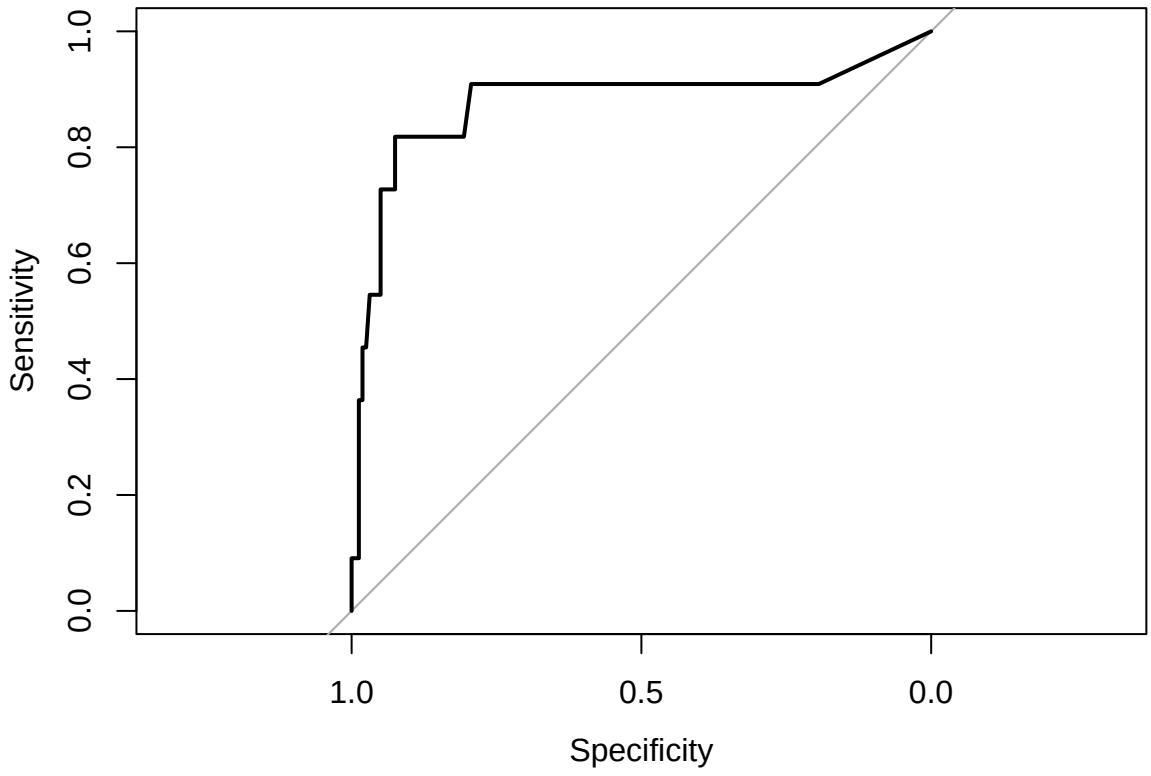
```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

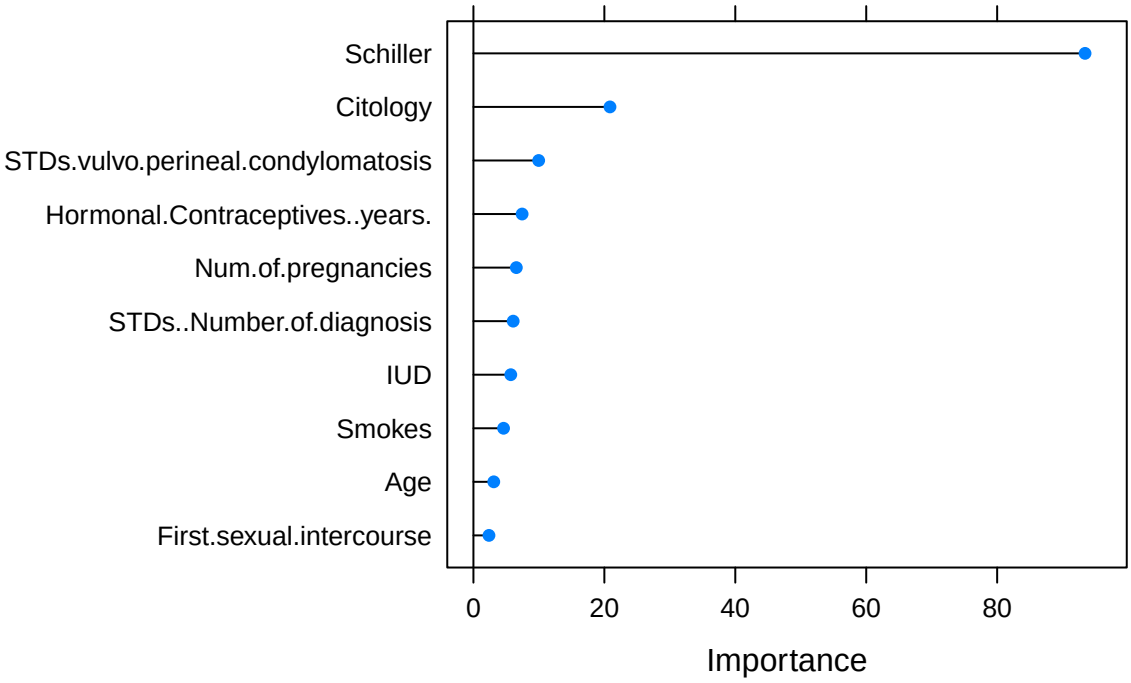
```
auc(roc_rf_rfe)
```

```
## Area under the curve: 0.8761
```

```
plot(roc_rf_rfe)
```




```
# var importance of final glmnet model
plot(varImp(rf_model_rfe, scale = FALSE), top = 10,
     main = "Important Factors for Predicting Cervical Cancer\n using Penalized Random Forest")
```



Threshold Investigation

```
threshold_df <- thresholds_cm(testResults_rf_rfe)
print(threshold_df)
```

```
##   Threshold TP FP  TN FN
## 1       0.1  9 20 140  2
## 2       0.2  9 15 145  2
## 3       0.3  8 12 148  3
## 4       0.4  8  8 152  3
## 5       0.5  8  8 152  3
## 6       0.6  8  8 152  3
## 7       0.7  8  8 152  3
## 8       0.8  8  8 152  3
## 9       0.9  6  6 154  5
```

Data_Ingestion.R

ruddysimonpour

2023-06-23

```
#install.packages("dplyr")
#install.packages("VIM")


library(ggplot2)
library("Hmisc")
```

```
##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:Hmisc':
##
##     src, summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(VIM)
```

```
## Loading required package: colorspace

## Loading required package: grid
```

```
## The legacy packages maptools, rgdal, and rgeos, underpinning this package
## will retire shortly. Please refer to R-spatial evolution reports on
## https://r-spatial.org/r/2023/05/15/evolution4.html for details.
## This package is now running under evolution status 0

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep
```

```
set.seed(007)
read_data <- function(x) {
  # will read data from input folder
  # used na.strings = "?" for the null values. The null values are stored as "?" in the dataset.
  cervical_data_raw <- read.csv(x, na.strings = "?")
  glimpse(cervical_data_raw)
  return(cervical_data_raw)
}


############################################################## check missing values
check_nulls <- function(df) {
  null_counts <- colSums(is.na(df))

  total_data_points <- nrow(df) * ncol(df)  # calculate the total number of data points

  total_nulls <- sum(null_counts) # calculate the total number of missing values

  column_percentage_null <- (null_counts / nrow(df)) * 100   # calculate the percentage of missing valu

  total_percentage_null <- (total_nulls / total_data_points) * 100   # calculate the percentage of miss

  null_counts_df <- data.frame(
    Column = names(null_counts),
    Nulls = as.numeric(null_counts),
    ColumnPercentage = column_percentage_null
  )

  null_counts_df <- rbind(null_counts_df, c("Total", "total_nulls", "total_percentage_null"))

  print(null_counts_df)
  return(null_counts_df)
}


################################################################ remove columns with more than 50% missing d
remove_cols <- function(df, threshold = 60) {
```

```r
  null_counts_df <- check_nulls(df)

  cols_to_remove <- null_counts_df$Column[null_counts_df$ColumnPercentage > threshold] #columns to remo

  df <- df[, !(names(df) %in% cols_to_remove)]

  return(df)
}


############################################################ impute missing data with median
impute_median <- function(df, columns) {
  for (col in columns) {
    df[[col]][is.na(df[[col]])] <- median(df[[col]], na.rm = TRUE) #ignoring missing values when calcul
  }
  return(df)
}


############################################################ impute missing data with mode for the binar
find_mode <- function(x) {
  u <- unique(x)
  tab <- tabulate(match(x, u))
  u[tab == max(tab)]
}


impute_mode <- function(df, columns) {
  for (col in columns) {
    df[[col]][is.na(df[[col]])] <- find_mode(df[[col]])
  }
  return(df)
}

############################################################ impute missing data using knn imputation
# Function to impute NA values with kNN imputation
impute_with_knn <- function(df, k = 6) {
  df_imputed <- kNN(df, k = k)  # Perform kNN imputation
  return(df_imputed)
}
```

Viz_EDA.R

ruddysimonpour

2023-06-23

```r
# ADS-503 Final Project: Cervical Cancer Biopsy Prediction
# Authors: Ruddy Simonpour & Shailja Somani

## distribution of the columns on separate plots
################################################################ Distribution of the columns
plot_distributions <- function(df, column_indices) {
  selected_cols <- names(df)[column_indices]

  for(col in selected_cols){
    p <- ggplot(df, aes_string(col)) + geom_histogram(bins = 30) + theme_minimal() +
        labs(title=col, x=NULL) +  theme(plot.title = element_text(hjust = 0.5))
    print(p)
  }
}

################################################################ Plot all histograms of a dataframe together
hist.df <- function(df) {
  par(mfrow=c(3,3))
  lapply(names(df), function(col) hist(df[, col], main=col, xlab="", ylab="", col="gray", breaks=10))
}



################################################################ Create boxplots for all features - helps vis
# Each plot will have 10 boxplots
boxplot.df <- function(df) {
  par(mfrow=c(2,5))
  for(i in 1:ncol(df)) {
    boxplot(df[[i]], main=colnames(df)[i], col="gray")
  }
}


################################################################ Scatter plot of columns
library(ggplot2)
create_scatterplot <- function(df, x_var, y_var, color_var, x_label, y_label, title, legend_title) {
  p <- ggplot(df, aes_string(x = x_var, y = y_var, color = color_var)) +
    geom_point(show.legend = TRUE) +
    labs(x = x_label, y = y_label, title = title, color = legend_title) +
    scale_color_gradient(low = "green", high = "red", na.value = "blue", guide = "legend") +
    theme_minimal() + theme(legend.position = "bottom")
  print(p)
```

```
}



########################################################## Heatmap of correlations

create_heatmap <- function(title, df) {
  cor_mat <- cor(df)
  cor_data <- reshape2::melt(cor_mat)

  # Create the heatmap using ggplot2
  heatmap <- ggplot(cor_data, aes(x = Var1, y = Var2, fill = value)) +
    geom_tile() +
    labs(title = title) +
    scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                         midpoint = 0, limits = c(-1, 1)) +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))

  return(heatmap)
}
```

Preprocessing.R

ruddysimonpour

2023-06-23

```r
# library(caret)

preprocess_data <- function(train_X, test_X) {
  preProcessData <- preProcess(train_X, method = c("center", "scale"))
  train_X <- predict(preProcessData, train_X)
  test_X <- predict(preProcessData, test_X)

  list(train_X = train_X, test_X = test_X)
}
```

Modeling.R

ruddysimonpour

2023-06-23

```
################################################ Neural Network Training Function
train_nnet_model <- function(train_X, train_y, ncol_train, cntrl) {
  nnetGrid <- expand.grid(decay = c(0, 0.01, .1), size = c(3, 7, 11, 13))
  set.seed(100)
  nnetTune <- caret::train(x = train_X, y = train_y,
                           method = "nnet",
                           tuneGrid = nnetGrid,
                           trControl = cntrl,
                           linout = FALSE, # FALSE => Classification task, TRUE => Regression task
                           trace = FALSE,
                           MaxNWts = 15 * ncol_train + 1,
                           maxit = 1000)
  return(nnetTune)
}

################################## Multivariate Adaptive Regression Splines (MARS)
train_mars_model <- function (train_X, train_y, nprune_range, cntrl) {
  set.seed(100)
  mars_model <- train(x = train_X, y = train_y,
                  method = "earth",
                  tuneGrid = expand.grid(degree = 1, nprune = nprune_range),
                  trControl = cntrl)
  return(mars_model)
}

######################################################## Support Vector Machine (SVM) (svmRadial)
train_svm_model <- function (train_X, train_y, tuneLength_range, cntrl) {
  set.seed(100)
  svmRTune <- train(x = train_X, y = train_y,
                    method = "svmRadial",
                    tuneLength = tuneLength_range,
                    trControl = cntrl)
  return(svmRTune)
}

################################################### Support Vector Machine (SVM) (svmPoly)
train_svm_poly <- function(train_X, train_y, cntrl){

  svm_grid <- expand.grid(degree = 1:2,
                          scale=c(0.01, 0.005, 0.001),
                          C=2^(-2:5))
```

```
  set.seed(100)

  svm_model <- train(x = train_X, y = train_y,
                     method = "svmPoly",
                     tuneGrid = svm_grid,
                     trControl = cntrl)

  return(svm_model)
}

########################################################## K-nearest Neighbor (kNN)
knn_model_train <- function(train_X, train_y, cntrl, k_range) {


  knnGrid <- expand.grid(k = k_range)

  set.seed(100)

  knnTune <- train(x = train_X,
                   y = train_y,
                   method = "knn",
                   tuneGrid = knnGrid,
                   trControl = cntrl)

  return(knnTune)
}

#################################################Random Forest (RF)
rf_model_train <- function(train_X, train_y, cntrl) {

  mtryGrid <- data.frame(mtry = floor(seq(10, ncol(train_X)/3, length = 10)))

  set.seed(100)

  rfTune <- train(x = train_X, y = train_y,
                  method = "rf",
                  tuneGrid = mtryGrid,
                  importance = TRUE,
                  trControl = cntrl)

  return(rfTune)
}

#################################################Logistic Regression Model
lr_model_train <- function(train_X, train_y, cntrl) {

  set.seed(100)

  # Train new model
  lrFit <- train(x = train_X,
                 y = train_y,
                 method = "glm",
                 metric = "ROC",
```

```
                trControl = cntrl)


  return(lrFit)
}

#########################################################LDA Model
lda_model_train <- function(train_X, train_y, cntrl) {

  set.seed(100)


  ldaFit <- train(x = train_X,
                  y = train_y,
                  method = "lda",
                  preProc = c("center","scale"),
                  metric = "ROC",
                  trControl = cntrl)
  return(ldaFit)

}

########################################Penalized Logistic Regression Model
glmn_model_train <- function(train_X, train_y, cntrl) {


  set.seed(100)

  glmnGrid <- expand.grid(alpha = c(0,  .1,  .2, .4, .6, .8, 1),
                          lambda = seq(.01, .2, length = 10))

  glmnFit <- train(x = train_X,
                   y = train_y,
                   method = "glmnet",
                   tuneGrid = glmnGrid,
                   metric = "ROC",
                   trControl = cntrl)
  return(glmnFit)

}

#########################################Nearest Shrunken Centroids Model
nsc_model_train <- function(train_X, train_y, cntrl) {

  set.seed(100)


  nscFit <- train(x = train_X,
                  y = train_y,
                  method = "pam",
                  tuneGrid = data.frame(threshold = seq(0, 25, length = 30)),
                  metric = "ROC",
                  trControl = cntrl)
```

```
  return(nscFit)

}



##############################################Prediction Results Function
get_prediction_results<- function(model, test_X, test_y) {

  set.seed(100)

  prediction <- predict(model, test_X, type = "prob")
  prediction_class <- ifelse(prediction[,2] > 0.5, 1, 0)
  results <- data.frame(
    observation = as.factor(test_y),
    prediction = as.factor(prediction_class),
    class_prob = prediction[,2]
  )

  return(results)

}


##############################################Function to calculate accuracy
get_accuracy <- function(model, test_X, test_y) {

  pred <- predict(model, newdata = test_X)
  acc <- postResample(pred, test_y)["Accuracy"]
  return(acc)
}


##################################################Recursive Feature Elimination
rf_rfe <- function(train_X, train_y) {

  set.seed(100)

  control <- rfeControl(functions = rfFuncs,
                        method = "cv",
                        number = 5,
                        verbose = FALSE)

  # Perform RFE
  rfe_result <- rfe(train_X, train_y,
                    sizes = c(1:ncol(train_X)),
                    rfeControl = control)

  # Access the selected features
  selected_features <- predictors(rfe_result)
  return(selected_features)

}
```

```r
##############################################Threshold Tuning
thresholds_cm <- function(results_df) {

  # Create an empty data frame to store the results
  threshold_df <- data.frame(Threshold = numeric(),
                             TP = numeric(),
                             FP = numeric(),
                             TN = numeric(),
                             FN = numeric(),
                             stringsAsFactors = FALSE)

  # Set the threshold increments
  threshold_increments <- seq(0.1, 0.9, by = 0.1)

  # Iterate over the threshold increments
  for (threshold in threshold_increments) {
    # Compute the confusion matrix using the given threshold
    confusion_matrix <- table(results_df$observation,
                              results_df$class_prob >= threshold)

    # Extract the TP, FP, TN, and FN from the confusion matrix
    # note: the AUC will not change based on the threshold, but the CM is useful from a business perspe
    TP <- confusion_matrix[2, 2]
    FP <- confusion_matrix[1, 2]
    TN <- confusion_matrix[1, 1]
    FN <- confusion_matrix[2, 1]

    # Append the results to the result data frame
    threshold_df <- rbind(threshold_df, data.frame(Threshold = threshold,
                                                   TP = TP,
                                                   FP = FP,
                                                   TN = TN,
                                                   FN = FN))

  }

  return(threshold_df)

}
```